

Treball de Fi de Grau

Grau en Enginyeria en Tecnologies Industrials

Circuit electrònic per controlar un vehicle seguidor de trajectòria seleccionable

MEMÒRIA

Autor: Adrià Albesa Querol
Directora: Rosa Rodríguez Montañés
Convocatòria: Juliol 2017



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

El projecte consisteix en dissenyar i construir el circuit de control del moviment d'un vehicle de tres rodes impulsades per dos motors de corrent continu. El sistema està basat en un microcontrolador PIC i el pilotatge es pot realitzar de forma manual, o bé basat en el color primari RGB de l'objecte acostat al vehicle a tal efecte.

La comunicació entre l'usuari i el vehicle es fa mitjançant una aplicació per dispositiu smartphone amb sistema operatiu Android dissenyada a partir de l'eina disponible online AppInventor2 del MIT (Massachusetts Institute of Technology). El suport de la comunicació es fa a través de senyals i protocol Bluetooth.

La programació del microcontrolador es fa en llenguatge d'alt nivell C i utilitzant l'entorn de programació i desenvolupament de projectes MPLAB del fabricant del microcontrolador Microchip.

La detecció dels components RGB del color de l'objecte acostat al vehicle es fa amb un sensor digital que es comunica amb el microcontrolador mitjançant un bus I2C.

SUMARI

Contenido

Resum	3
1. PREFACI	6
1.1 Origen del projecte	6
1.2 Motivació	6
1.3 Requeriments previs	7
2. Introducció	8
2.1 Objectius del projecte	8
2.2 Abast del projecte	8
3. Microcontrolador PIC 16F690 de Microchip	9
3.1 I2C (Synchronous Serial Port).....	12
3.2 Interrupcions	14
3.2.1 Interrupcions habilitades	14
3.2.2 Timer 0	16
3.3 UART.....	17
3.3.1 Transmissió	18
3.3.2 Recepció.....	19
4. Eines de programació	21
4.1 MPLAB IDE.....	21
4.2 PICKIT 3	22
4.3 PICKIT 3 connectat al PIC16F690	22
5. Construcció del vehicle	24
6. Circuit electrònic	25
7. Mòdul Bluetooth	27
8. HC-SR04	28
8.1 Funcionament	28
9. Reguladors de tensió	32
9.1 Regulador 7805	32
9.2 Regulador LM1117	33
10. Motor	34

10.1	Motor elèctric	34
10.2	PWM programat.....	35
10.3	PWM Pulse Width Modulation.....	37
11.	Amplificador de corrent L293D	40
12.	Canviador de nivell lògic.....	42
13.	Sensor de llum	43
13.1	Condicions de Funcionament.....	44
13.2	I2C	45
13.2.1	Funcionament	45
13.3	Interrupció INT	49
13.4	RGB.....	49
13.5	Codi RGB.....	51
14.	Aplicació Mòbil	53
14.1	Introducció a l'aplicació	53
14.2	Connectivitat Bluetooth	57
14.3	Control vehicle	57
14.3.1	Sliders.....	57
14.3.2	Botons	66
14.4	Dades Sensor de llum.....	67
15.	Aplicació del sensor	70
15.1	Control del vehicle pel color	70
15.2	Relació entre color i dada enviada	76
16.	Programa	77
17.	Pressupost	83
18.	Impacte mediambiental	86
19.	Conclusions.....	87
20.	Agraïments.....	88
21.	Bibliografia	89

1. PREFACI

1.1 Origen del projecte

L'origen del projecte neix per l'interès de l'autor a especialitzar-se en el món de l'electrònica i poder prendre contacte per primer cop amb un microcontrolador i dissenyar un sistema basat en aquest i treballar amb un sensor digital intel·ligent de llum que capta informació sobre el color dels objectes.

1.2 Motivació

En aquest treball de final de grau, l'autor ha volgut realitzar un treball dinàmic i experimental on es combinessin l'electrònica i la informàtica.

El projecte desenvolupat és complet en el sentit que requereix una part de Hardware, que inclourà tot el circuit creat al voltant del Microprocessador que aconsegueix moure les rodes del vehicle, i una part de Software, que serà el programa creat en llenguatge C que indicarà al circuit tot allò que ha de fer.

La part d'electrònica obliga a qualsevol enginyer a seguir una metodologia per concebre i dissenyar (especificacions, concepció i validació) el circuit electrònic que ha de fer moure el vehicle, per tal de provocar el mínim d'errors, ja que el fet de provocar errors en un circuit electrònic pot suposar la inutilització dels components i de pas pot provocar l'augment del cost del projecte. Tant el temps de desenvolupament del producte com el cost d'aquest, són uns elements fonamentals de la tasca d'un enginyer.

Pel que fa a la part relacionada a la informàtica, és necessari saber comunicar-se amb el microprocessador per tal d'activar els senyals que es volen utilitzar per tal que el circuit realitzi la seva tasca. En aquest projecte, s'han d'utilitzar dos llenguatges desconeguts inicialment per l'autor; són el llenguatge C i el llenguatge de programació d'aplicacions per Android.

Per tot això, combinant aquestes dues branques de l'enginyeria, l'autor ha volgut realitzar un projecte de disseny d'un sistema electrònic que permet posar en pràctica els coneixements de l'electrònica d'una forma creativa, experimental i dinàmica.

1.3 Requeriments previs

El punt central del projecte està basat en el funcionament d'un microprocessador i en la forma de comunicar-se amb ell els perifèrics que controla. Per assolir els objectius del projecte, s'ha hagut de analitzar i conèixer a fons el PIC 16F690 de Microchip a partir del seu manual, ja que allà el fabricant ens explica tota la seva funcionalitat, com les seves interrupcions i els seus perifèrics.

En el període de realització d'aquest treball, l'autor ha enriquit i desenvolupat els seus coneixements sobre llenguatges de programació amb el llenguatge C, un llenguatge molt utilitzat en la indústria de microcontroladors .

Per últim, cal destacar la creació d'una aplicació per a mòbil per tal de controlar el sensor amb el vehicle via Bluetooth, fet que ha obligat a conèixer com es crea aquest tipus d'aplicacions.

2. Introducció

En aquest apartat s'explicaran els objectius i l'abast del projecte. Com s'ha explicat al resum, consisteix en la construcció d'un circuit electrònic per tal de controlar el moviment d'un vehicle de tres rodes mogudes amb dos motors DC, així que s'hauran de complir una sèrie d'objectius per tal de finalitzar amb èxit el projecte.

2.1 Objectius del projecte

L'objectiu del projecte és dissenyar i construir el sistema electrònic basat en un microcontrolador PIC que permeti controlar el moviment d'un vehicle de tres rodes de tal manera que l'usuari pugui decidir la direcció i la velocitat en tot moment.

El pilotatge es podrà realitzar de forma manual (escollint velocitat i direcció), o bé basat en l'execució d'ordres predeterminades i seleccionades a partir del color primari RGB d'un objecte acostat al vehicle.

La comunicació entre l'usuari i el vehicle es farà mitjançant una aplicació per dispositiu smartphone amb sistema operatiu Android i es dissenyarà a partir de l'eina disponible online AppInventor2 del MIT (Massachusetts Institute of Technology). La comunicació es farà a través de senyals i protocol Bluetooth. La programació del microcontrolador es farà en llenguatge d'alt nivell C.

La detecció dels components RGB del color de l'objecte acostat al vehicle es farà amb un sensor digital que es comunicarà amb el microcontrolador mitjançant un bus I2C.

2.2 Abast del projecte

El projecte es centra a construir el circuit electrònic de control del vehicle, basat en un microcontrolador i adjuntat sobre la seva estructura, que pugui controlar el moviment dels dos motors de corrent continu a través d'un telèfon mòbil i a través del color que capti el sensor de llum adjuntat. El control del moviment es farà seleccionant la potència transmesa a cada motor. No hi haurà realimentació sobre la velocitat obtinguda.

Per realitzar-ho, es plantejarà la utilització del material disponible al laboratori docent del departament d'Enginyeria Electrònica de l'ETSEIB.

3. Microcontrolador PIC 16F690 de Microchip

El microcontrolador utilitzat per controlar el vehicle ha estat el PIC 16F690 de la marca Microchip [3]. El dispositiu pertany a la família dels PICs, de l'anglès Peripheral Interface Microcontroller.

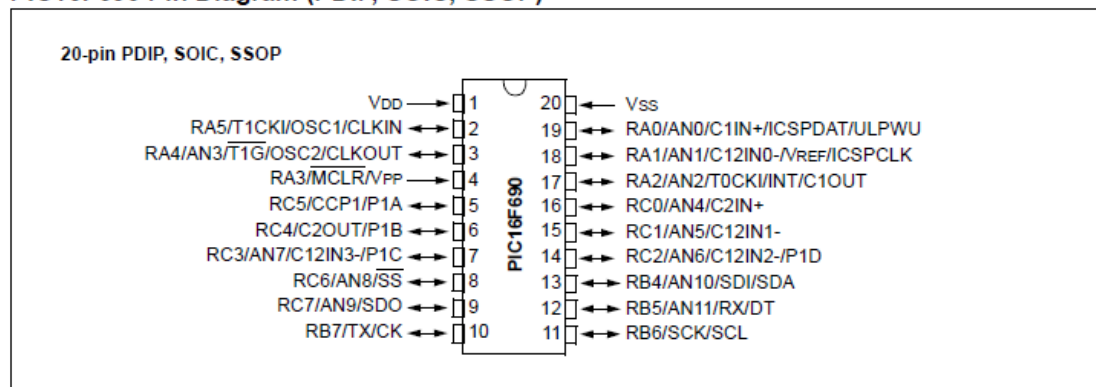
Aquest microcontrolador té les característiques de la Figura 3.1.

Device	Program Memory	Data Memory		I/O	10-bit A/D (ch)	Comparators	Timers 8/16-bit	SSP	ECCP+	EUSART
	Flash (words)	SRAM (bytes)	EEPROM (bytes)							
PIC16F631	1024	64	128	18	—	2	1/1	No	No	No
PIC16F677	2048	128	256	18	12	2	1/1	Yes	No	No
PIC16F685	4096	256	256	18	12	2	2/1	No	Yes	No
PIC16F687	2048	128	256	18	12	2	1/1	Yes	No	Yes
PIC16F689	4096	256	256	18	12	2	1/1	Yes	No	Yes
PIC16F690	4096	256	256	18	12	2	2/1	Yes	Yes	Yes

Figura 3.1- Característiques PIC16 F690

Les característiques més rellevants del PIC16F690 per a la realització d'aquest projecte, són la seva Data Memory (RAM) de 256 bytes, els seus 18 pins d'entrada/sortida i els timers (0,1,2), la EUSART (UART), el mòdul PWM i el bloc SSP, els quals són perifèrics.

Aquest PIC té 2 pins d'alimentació i 18 pins d'entrada i sortida repartits en els Ports A, B i C. Al Port A hi ha 4 pins que són d'entrada/sortida i un pin que només és d'entrada (RA3), al Port B hi ha 4 pins d'entrada/sortida i al Port C hi ha 7 pins d'entrada/sortida. D'aquests pins, hi ha alguns que tenen una connexió directa a diferents mòduls que tenen la capacitat de connectar amb determinats perifèrics. Aquests pins estan indicats a Figura 3.2, però seran comentats més endavant per explicar cada part del funcionament del vehicle.

PIC16F690 Pin Diagram (PDIP, SOIC, SSOP)**Figura 3.2- Esquema pinout del PIC16F690**

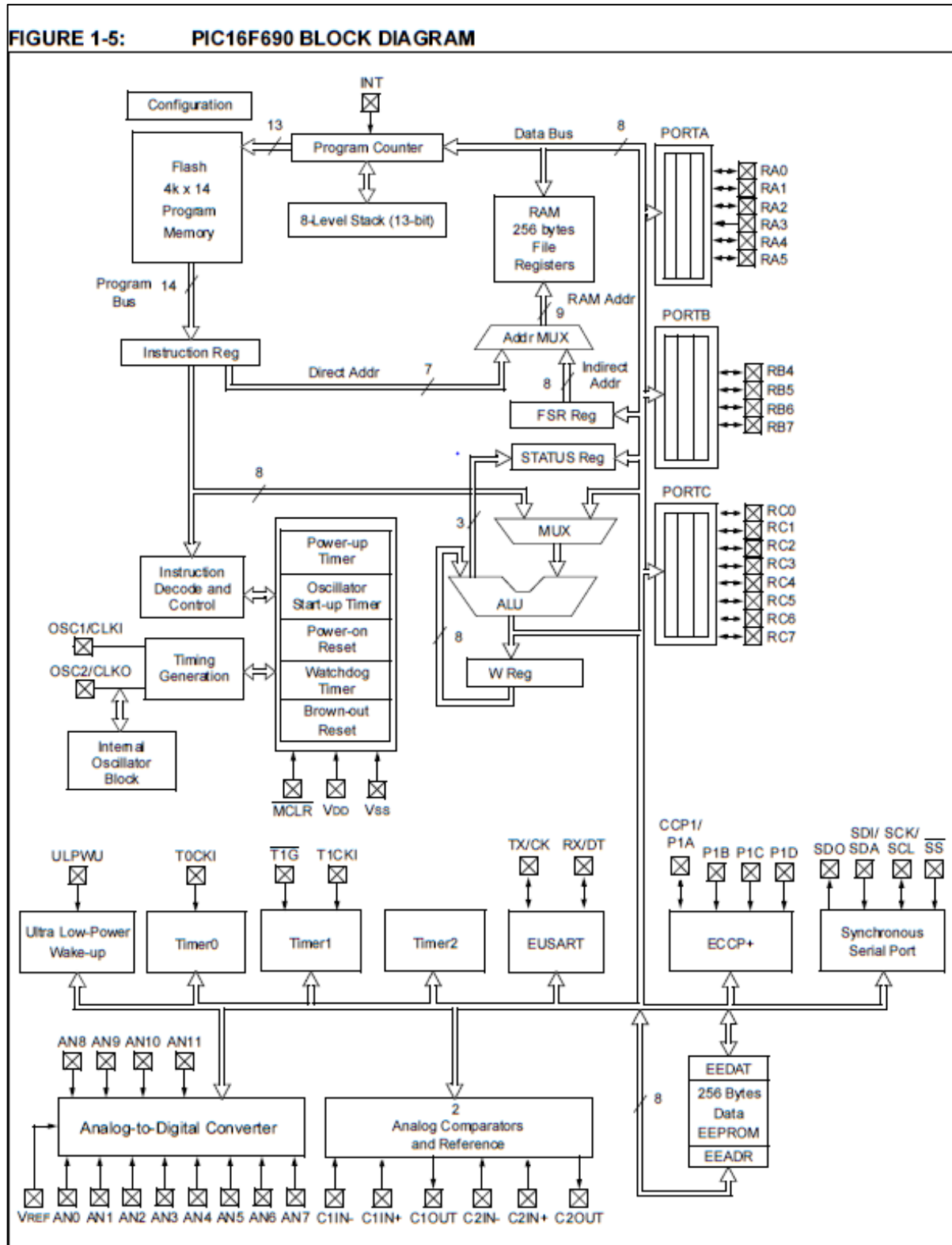


Figura 3.3- Diagrama de blocs del PIC

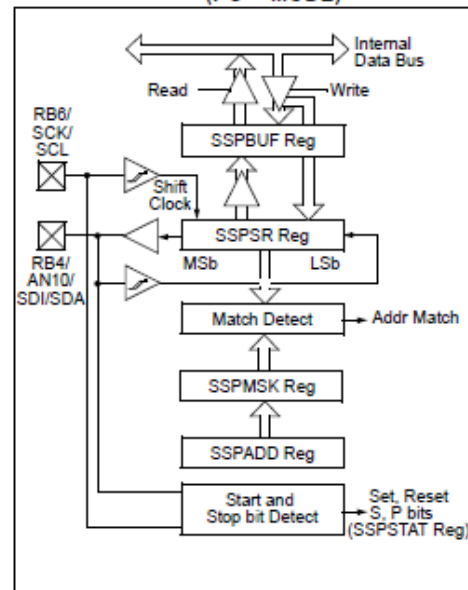
La Figura 3.3 mostra totes les parts que té el microcontrolador i la interconnexió entre elles. Com a elements a destacar, s'observa que la Flash Program Memory (ROM) és de 4k x 14 i ha d'allotjar el programa que controlarà el vehicle. En la memòria RAM, que conté 256 bytes, és on s'emmagatzemaran les dades processades en el projecte. En el diagrama s'observa com aquests blocs interactuen amb el Program Counter i la seva pila de 8 nivells i de 13 bits d'amplada de paraula.

A més, es diferencia clarament el Bus de dades (Data Bus) de 8 bits que interactua amb els pins entrada/sortida Ports A, B i C i amb d'altres elements com els Timers. En aquest projecte, s'utilitzen el Timer 0 de 8 bits i el Timer 1 de 16 bits, l'EUSART (UART) on es veuen els pins RX (recepció) i Tx (transmissió), i el mòdul Synchronous Serial Port amb els senyals SDA (Signal Data) i SCL (Signal Clock) que es connectaran pel bus I2C cap al sensor de llum extern.

Els següents apartats es presenten individualment els mòduls del microcontrolador utilitzats en aquest projecte i es comenten les seves característiques més importants.

3.1 I2C (Synchronous Serial Port)

El microcontrolador es connecta al sensor de llum utilitzant el bus I2C que pertany al bloc de SSP (Synchronous Serial Port) amb els senyals SDA (Signal Data) i SCL (Signal Clock). Com es mostra a la Figura 3.4 aquests dos senyals surten, segons indica el manual del PIC 16F690, del pin RB4 (SDA) i del pin RB6 (SCL) i transmeten o reben les dades del SDA al registre SSP Shift Register començant pel LSB (Least Significant Byte) i acabant pel MSB (Most Significant Byte). En aquest registre és on es realitzen les operacions de lectura o escriptura de dades que es transporten a través del bus SSPBUF. Aquest bus s'explica més profundament a l'apartat I2C del sensor de llum.

FIGURE 13-7: SSP BLOCK DIAGRAM (I²C™ MODE)**Figura 3.4- Diagrama de blocs del SSP**

En el protocol que utilitza aquest bus, el microcontrolador s'anomena Mestre, ja que és qui dóna les ordres a l'Esclau, que serà el sensor. La particularitat que té aquest microcontrolador és que està completament preparat per a funcionar en mode "Esclau" i que si es vol treballar en el mode Mestre s'ha de programar el software per a que realitzi aquesta funció ja que no té implementat la totalitat del protocol.

La manera de realitzar aquesta implementació del mode mestre segons el manual del PIC 16F690 és definint els pins RB4 (SDA) i el pin RB6 (SCL) com a entrades, és a dir, que aquests pins estan en alta impedància sempre que el mestre no envii dades cap a l'exterior o bé n'envii un 1. Aquest estat és molt útil ja que el que fa és evitar que mentre no hi hagi connexió entre el Mestre i l'Esclau aquests dos pins no puguin prendre valors aleatoris i se'ls imposi un senyal digital de '1'. Contràriament, sempre que el mestre vulgui enviar un 0 cap als pins del bus I2C, simplement s'haurà de configurar els pins com a sortides. Automàticament, els pins seran connectats a nivell lògic baix independentment del valor escrit en el port, segons indica el datasheet del microcontrolador.

Segons el manual del PIC 16F690 aquests pins estan connectats a un pull-up, és a dir, a una resistència i a una tensió de 5V. Per tant, com s'ha explicat, si s'introdueix un '1' a aquests pins, no hi circularà corrent.

Aquest protocol apartat s'amplia a l'apartat del **Sensor de llum** on s'explica tot el protocol I2C.

3.2 Interrupcions

3.2.1 Interrupcions habilitades

El microcontrolador PIC 16F690 té varies fonts possibles d'interrupcions:

- Interrupció externa RA2/INT
- TMR0 interrupció de desbordament
- Canvi d'interrupcions PORTA/PORTB
- 2 comparadors d'interrupcions
- Interrupció A/D
- Interrupció de desbordament Timer 1
- Interrupció de desbordament Timer 2
- Interrupció d'escriptura de dades a la EEPROM
- Interrupció de Fail-Safe Clock Monitor
- Interrupció CCP millorada
- Interrupcions de transmissió i recepció de l'EUSART

Les interrupcions utilitzades al projecte són les de recepció de l'EUSART, la interrupció de canvi PORTA/ PORTB, la interrupció externa del pin RA2 i la interrupció del Timer 0. Per poder activar qualsevol de les interrupcions s'habiliten al programa el senyal Global Interrupt Enable (GIE), que activa totes les interrupcions sense emascarar i el senyal Peripheral Interrupt Enable Bit PEIE, que activa les interrupcions dels perifèrics. Dels bits d'habilitació del PEIE només s'activarà al programa el PIR1.

Un cop tenim les interrupcions no emascarades i les dels perifèrics activades, només falta activar els bits dels registres que activen cada interrupció en particular. Cada cop que es compleixi la condició d'una de les interrupcions activades, s'indicarà amb la posada a nivell alt del flag corresponent.

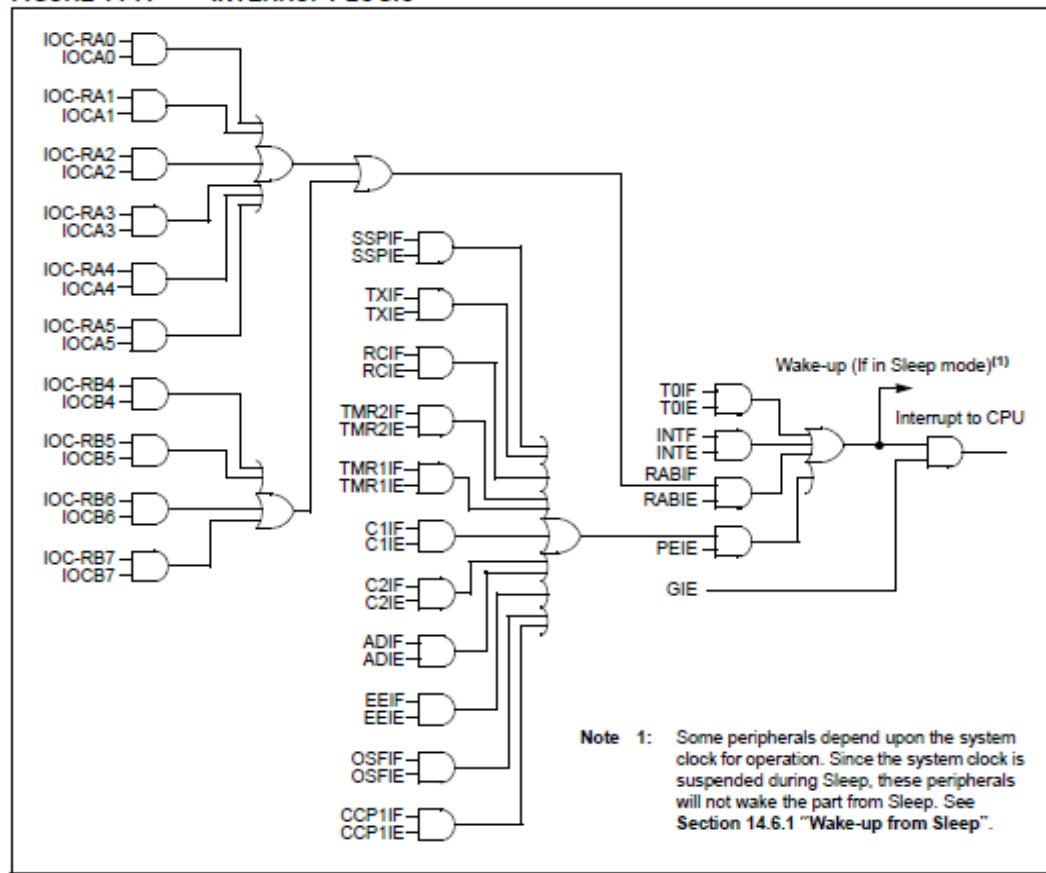
A la interrupció del perifèric UART (Universal Asynchronous Receiver Transmitter) i més en concret la recepció de la UART hi ha un bit d'activació que s'anomena Eusart Receive Interrupt bit (RCIE). Com s'observa al programa [ANNEX], el servei d'interrupció corresponent només s'executarà quan el flag de la interrupció de la recepció de la UART es mantingui a 1.

En aquest projecte, per detectar la distància a un objecte extern s'utilitza el sensor d'ultrasons HC-SR05. Aquest sensor indicarà un perill de xoc quan la distància sigui menor que un cert valor de seguretat. Aquesta detecció es farà a través d'un canvi en el nivell lògic d'un senyal concret i s'ha utilitzat la interrupció de canvi de PORTA/PORTB que s'activa amb el pin de la interrupció IOCA5 a '1', el bit RABIE del registre INTCON a '1'. El flag corresponent és el bit RABIF del mateix registre. En acabar la interrupció, cal posar a '0' el flag corresponent, bit RABIF (Figura 3.7).

A l'hora d'aplicar una interrupció externa al pin RA2 només cal activar el bit INTE (registre INTCON), per indicar que es produirà una interrupció externa, i el bit INTF per entrar al servei d'interrupció externa. Per netejar-la només cal posar a '0' el bit INTF. Aquesta interrupció s'utilitza per activar el bus I2C que connecta el PIC 16F690 amb el sensor de llum APDS-9960.

A la Figura 3.5 s'observa com la combinació explicada es correspon en el diagrama lògic de les interrupcions:

FIGURE 14-7: INTERRUPT LOGIC

**Figura 3.5- Blocs de les interrupcions del PIC16F690**

3.2.2 Timer 0

La Figura 3.6 il·lustra la possibilitat de selecció del rellotge que activa el Timer 0. Com en aquest projecte tota la part del WDT (Watch Dog Timer) està desactivada i llavors, segons el manual, només s'ha d'activar el bit T0CS, posant el registre Option Register a '0'.

Seguidament, per activar la interrupció per desbordament del Timer 0 cal activar el bit T0IE a '1' (registre INTCON) i per a que s'entri dins el servei de la interrupció corresponent s'ha de comprovar que el bit T0IF està a '1'. Un cop finalitzada la interrupció, s'ha de netejar només el bit T0IF.

El valor del registre TMR0 indica el comptatge d'aquest i, coneixent la freqüència del seu rellotge, és un indicador del temps transcorregut des de la darrera interrupció per

desbordament del mateix (coneixent el valor carregat inicialment. Amb aquesta finalitat s'ha utilitzat el Timer 0 en aquest projecte.

A l'apartat de Motor s'explica com s'ha programat el Timer 0 per a fer funcionar els motors del vehicle.

FIGURE 5-1: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER

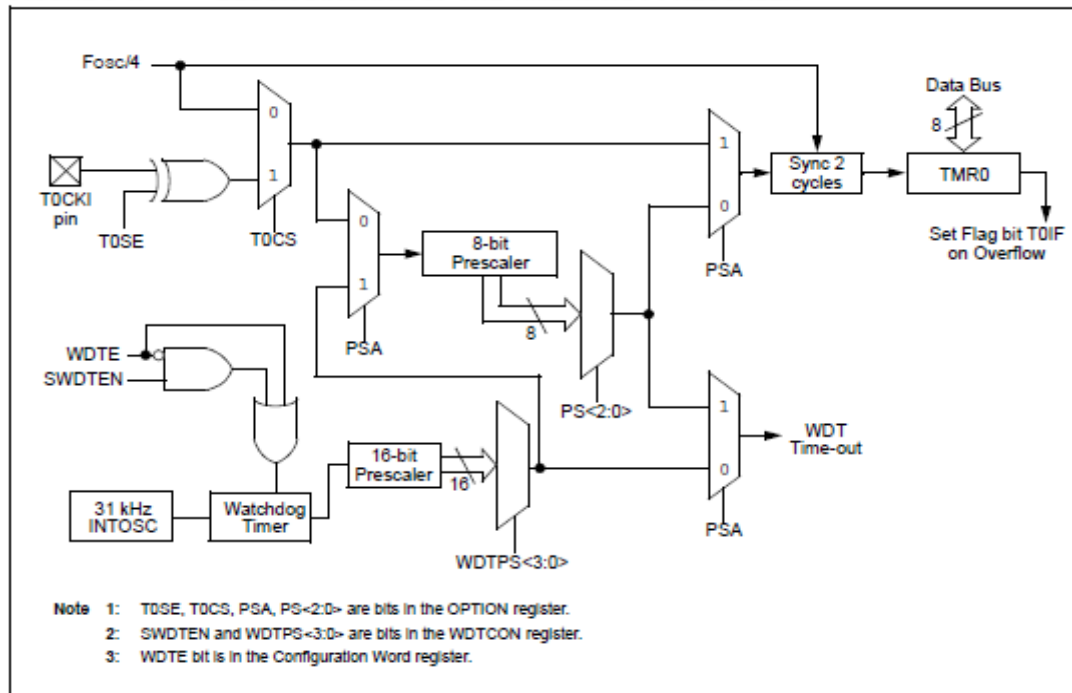


Figura 3.6- Diagrama de blocs Timer 0

REGISTER 2-3: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TOIE	INTE	RABIE ^(1,3)	TOIF ⁽²⁾	INTF	RABIF
bit 7							bit 0

Figura 3.7- Registre INTCON

3.3 UART

La Universal Asynchronous Receiver Transmitter (UART) és un perifèric que forma part del PIC 16F690 que permet la transmissió i la recepció de dades en format sèrie d'un microcontrolador amb un circuit integrat exterior. En aquest cas, amb el mòdul Bluetooth.

La UART és la part asíncrona del mòdul Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART), és a dir, el senyal de recepció i el senyal de transmissió de dades tenen cadascun un rellotge (CLK) independent entre sí.

Per tal d'activar aquest perifèric s'ha de configurar que el microcontrolador tingui un senyal de sortida (Tx) i un senyal d'entrada (Rx).

3.3.1 Transmissió

A la figura 3.8 es mostra un exemple genèric de la transmissió de dades en format sèrie a través del mòdul UART. S'observa com primer de tot, es registra la paraula que es vol enviar al Registre de transmissió (TXREG) i després s'activa el bus de transmissió i es produeix un canvi en el senyal de transmissió, que passa de 1 a 0, indicant que al següent flanc de pujada del rellotge BRG s'inicia la transmissió de la paraula bit a bit amb un bit Start que passa a 0.

Aquesta transmissió sempre segueix el mateix ordre (Figura 3.8), del bit menys significatiu (LSB) al bit més significatiu (MSB). Un cop s'ha transmes l'últim bit, s'activa el bit d'aturada (Stop bit) i al següent flanc de baixada del rellotge Baud-Rate Generator (BRG), el senyal de canvi de transmissió torna al valor de 1 i el bit d'aturada (Stop bit) es desactiva quan es produeix el següent flanc de pujada del rellotge BRG.

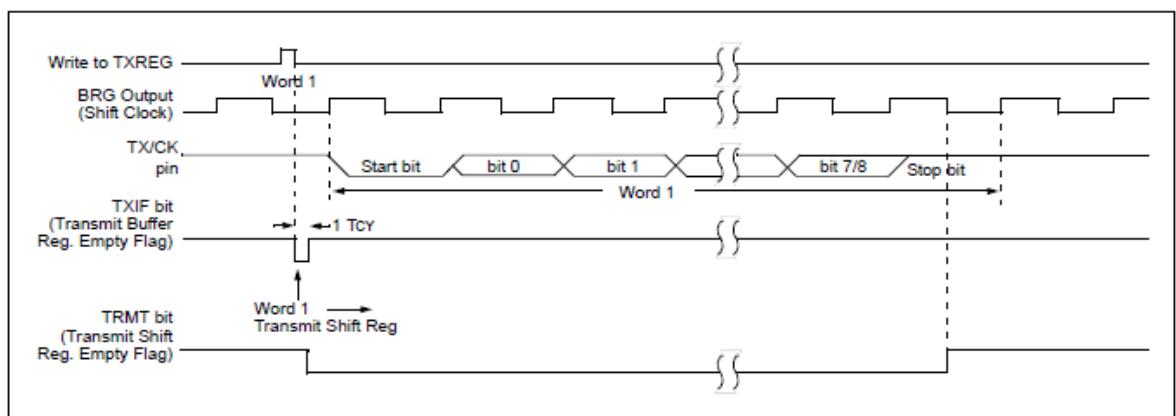


Figura 3.8- Transmissió TX

3.3.2 Recepció

A la Figura 3.9 es mostra un exemple general de recepció de dades a través del mòdul UART. Primer de tot, es rep la primera paraula i, un cop s'ha enviat el bit d'aturada (Stop bit), s'activa el bus de registres de recepció i s'activa la interrupció de recepció indicada pel seu flag RCIF. El senyal CREN és el que habilita la recepció i com s'observa, està activat a 1.

A continuació arriba la segona paraula i després la tercera paraula i es produeix la mateixa conversió de bits. En aquesta figura, en concret, es mostra un fenomen anomenat Overrun, que es produeix quan a causa de la cua de dades que hi ha, no es pot convertir la tercera paraula. Aleshores s'activa el OERR bit que neteja el senyal CREN i atura la recepció. Per tant, només es llegeixen al bus de recepció de lectura les dues primeres paraules.

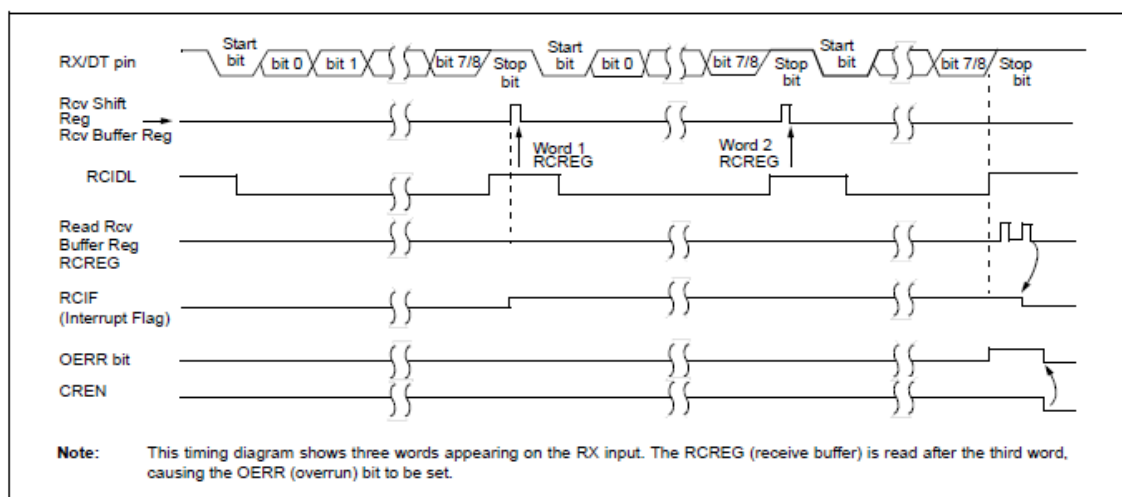


Figura 3.9- Recepció RX

Es mostra a la Figura 3.10 la visualització per oscil·loscopi d'una recepció sèrie experimental realitzada en aquest projecte entre el mòdul Bluetooth HC-05 i el microcontrolador. A la figura, es mostra que s'està transmetent una dada i es pot apreciar com aquesta recepció comença amb un bit de Start, que posa el senyal a 0 i seguidament envia tot el byte des del bit menys significatiu fins a al bit més significatiu. Aquest bit que s'envia és el 0101001, que correspon al número 81 en decimal.

Un altre fet destacat d'aquesta imatge és que es pot veure que el període d'enviament de cada bit es correspon amb el teòric, ja que si la velocitat d'enviament de cada bit/s és de

9600, el període de cada bit és de $104\ \mu\text{s}$. A la imatge es veu que per cada $250\ \mu\text{s}$, hi ha més de 2 bits.

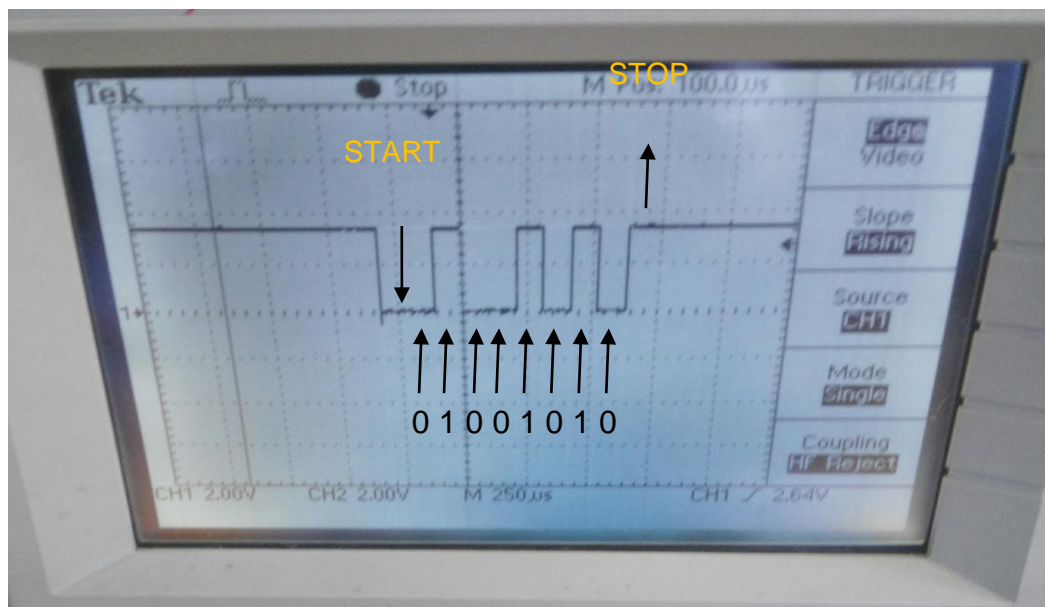


Figura 3.10- Recepció RX d'un byte des de l'oscil·loscopi

4. Eines de programació

4.1 MPLAB IDE

És un software utilitzat per desenvolupar aplicacions per a microcontroladors de Microchip i controladors digitals.

Per tal de desenvolupar l'aplicació per al funcionament del vehicle s'ha hagut de crear un projecte amb l'eina "Project Wizard" on s'ha definit el microcontrolador PIC 16F690, el compilador CX8, que permet transformar un codi en llenguatge C en llenguatge ensamblador, i les llibreries de Microchip per a treballar amb el PIC.

Els projectes de MPLAB permeten treballar amb varis fitxers principals "Source Files" i amb tots els seus fitxers afegits "Header Files". Com es veu a la Figura 4.1, el software mostra quin és el fitxer al qual s'està treballant (cotxe.c) i a quins fitxers està connectat

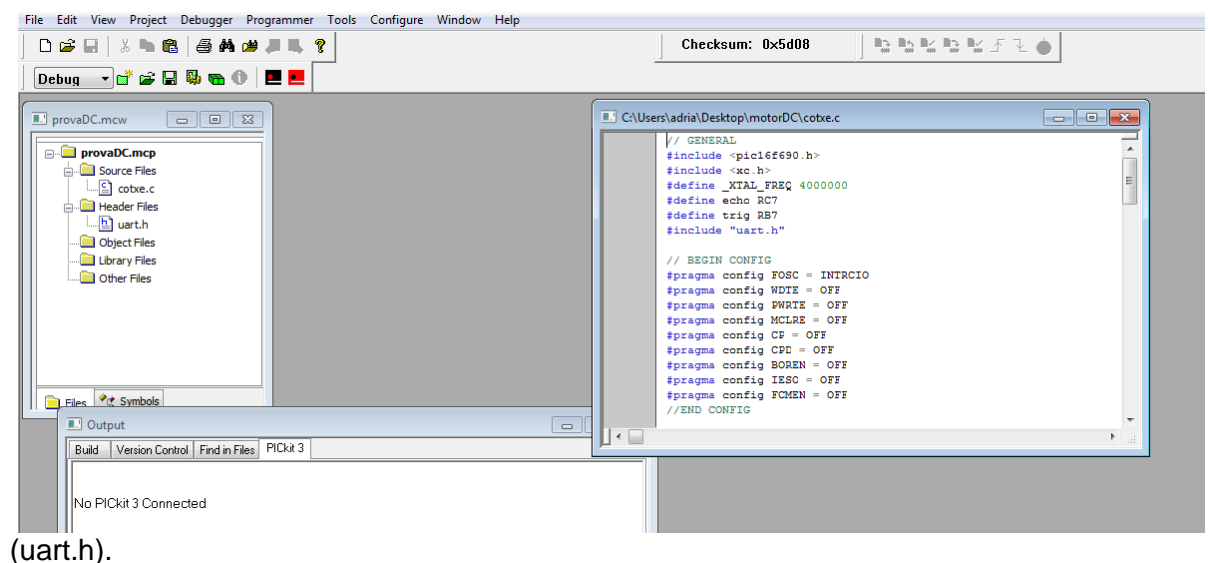


Figura 4.1- Interfície MPLAB

Aquest software combinat amb el compilador de llenguatge C CX8, ha permès comprovar que el codi de l'aplicació no contingués cap error amb l'eina "BUILD" abans de programar-lo al xip. Per a programar el PIC 16F690 s'ha utilitzat un altre element, hardware en aquest cas, el PICKIT 3.

4.2 PICKIT 3

El programador utilitzat per programar l'aplicació al microcontrolador PIC 16F690 és el PICKIT 3 [4], que també està fabricat per MICROCHIP.

El programador es caracteritza per ser compatible amb el software MPLAB IDE, ja que té una sortida USB per a connectar-lo a l'ordinador. Altres característiques d'aquest programador són suportar tensions de 2 Volts, normalment d'entre 2 i 6 volts, la capacitat de llegir i escriure sobre la ROM i la RAM del microcontrolador a més de poder esborrar la memòria mitjançant l'opció de verificació.

És important destacar la seva capacitat de suportar sobretensions i curtcircuits, ja que una de les aplicacions serà engegar un motor.

Per tal de connectar el programador al software s'ha de configurar que li arribin 5 volts i que el programador és el PICKIT 3 a l'opció PROGRAMMER/SETTINGS/POWER, sinó MPLAB no alimenta el programador.

4.3 PICKIT 3 connectat al PIC16F690

Per connectar els dos elements s'ha de seguir una determinada connexió indicada en el mateix manual del PICKIT 3.

Com mostra la Figura 4.2, s'han de connectar 5 pins, on el pin 1 correspon a la fletxa que hi ha dibuixada al programador i la connexió és la següent:

- El pin 1 correspon al MCLR/Vpp que correspon al pin 4 del microcontrolador (RA3)
- El pin 2 correspon al VDD de 5 volts que correspon al pin 1 del microcontrolador.
- El pin 3 correspon al Ground que seria el pin 20 del microcontrolador.
- El pin 4 correspon al ICSPDAT que correspon al pin 19 del microcontrolador (RA0).
- El pin 5 correspon al ICSPCLK que correspon al pin 18 del microcontrolador (RA1).

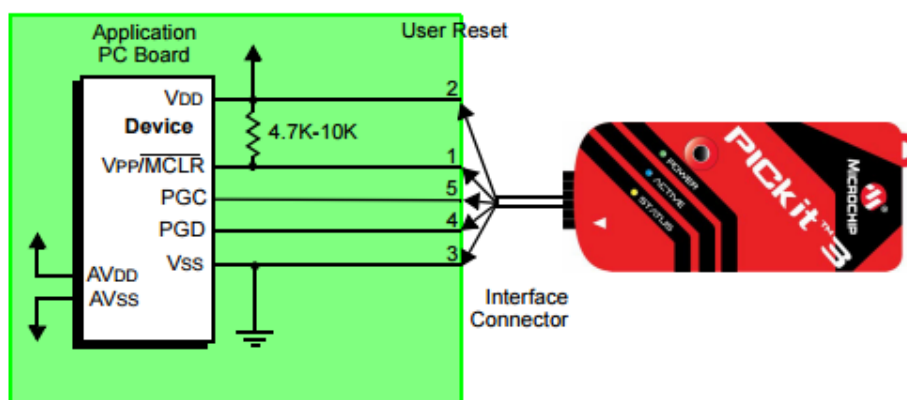


Figura 4.2- Connexions PICKIT 3

5. Construcció del vehicle

A l'hora d'escollir el vehicle utilitzat en aquest projecte s'ha buscat un model senzill ,i lleuger, ja que l'objecte principal d'aquest projecte no és tant muntar el vehicle, sinó dissenyar i construir el circuit electrònic que permeti controlar els moviments d'aquest vehicle. Per tant, s'ha construït un vehicle de 3 rodes **[13]** on dues d'elles necessiten un motor per moure's i l'altra s'adapta a la direcció que marquin les altres dues.

El vehicle, com es veu a la Figura 5.1, es basa en una estructura de metacrilat, a la qual se li han afegit tots els altres components com els motors i les rodes. Cada motor s'ha hagut de subjectar a la placa amb dos sostenidors de metacrilat amb dos cargols i dues rosques i la roda lliure ha estat fixada amb 4 espaiadors i 8 cargols.

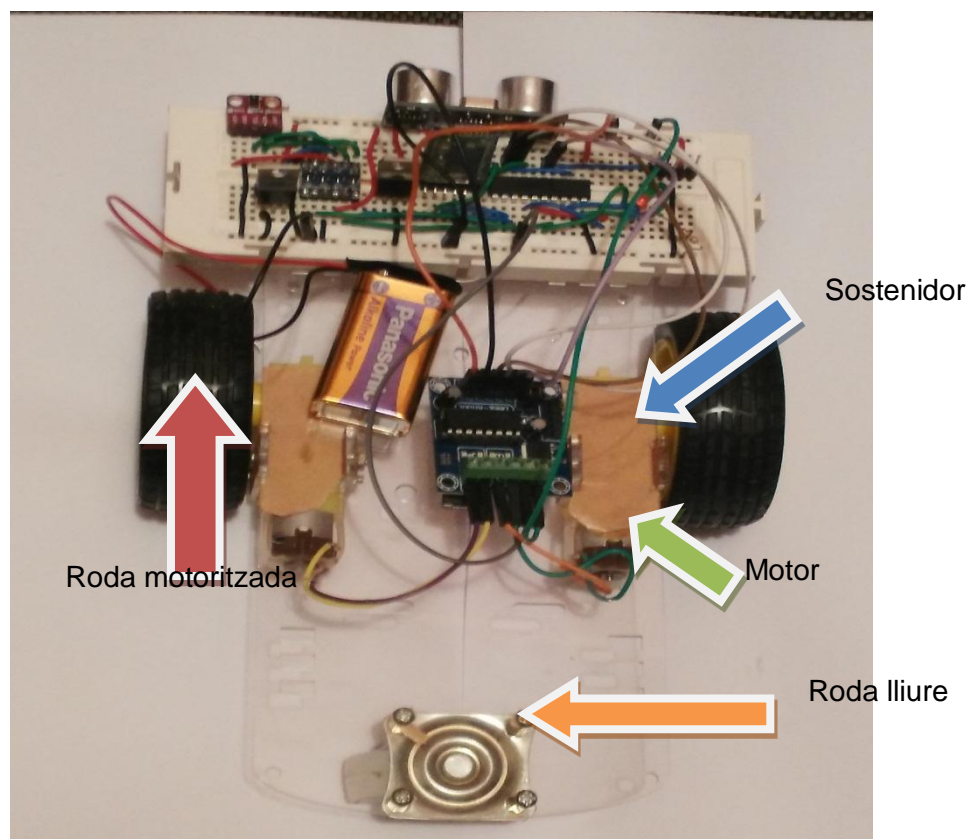


Figura 5.1- Elements del vehicle

6. Circuit electrònic

La Figura 6.1 mostra una fotografia del circuit electrònic muntat en aquest projecte i la figura 6.2 mostra el seu esquema elèctric realitzat amb el software OrCAD. S'observa com el PIC 16F690 és al centre de la imatge amb cada pin amb el nom del senyal que hi intervé segons el programa.

El circuit està format de varis components; a l'esquerra es troben els pins del programador PICKIT 3 i el sensor d'ultrasons HC-SR04; a la dreta es troben (de dalt a baix) el canviador de nivell amb el sensor de llum APDS-9960, el regulador de tensió 1117, el mòdul Bluetooth HC-05 i els dos motors; a dalt es troba la pila de 9V i el regulador de tensió 7805, per a poder treballar a 5V, i a sota trobem tres leds de color vermell, verd i blau amb una resistència de 220 Ohms.

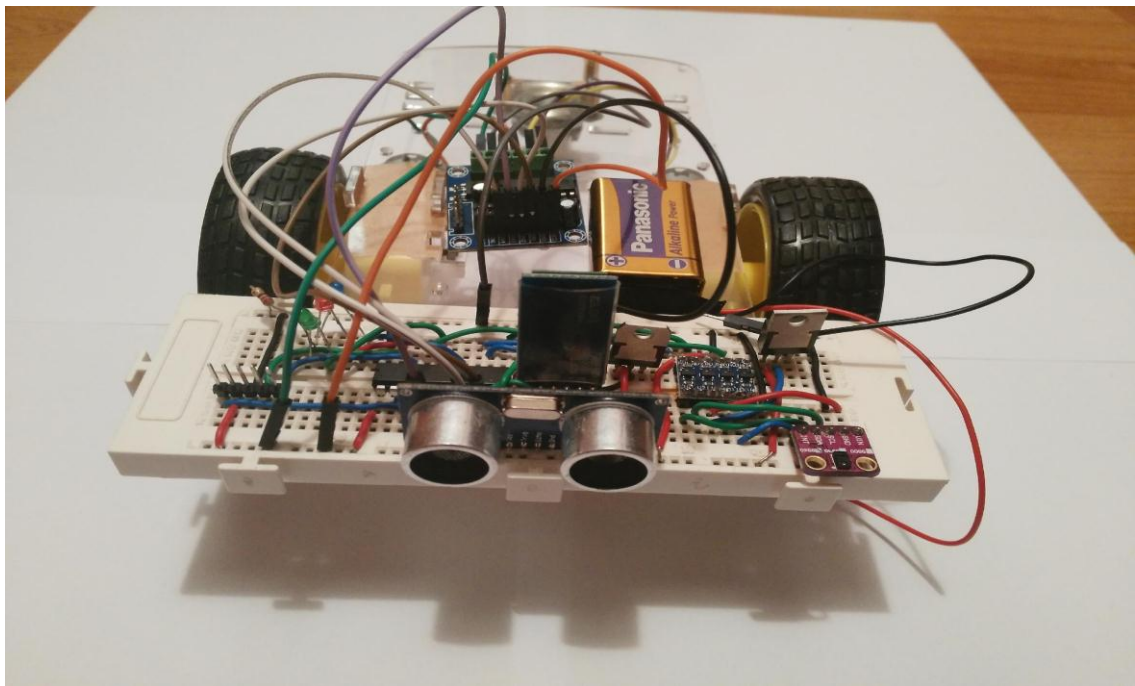


Figura 6.1- Circuit electrònic integrat al vehicle

A continuació s'explicarà les característiques i la funció dins del projecte de cada un d'aquests components.

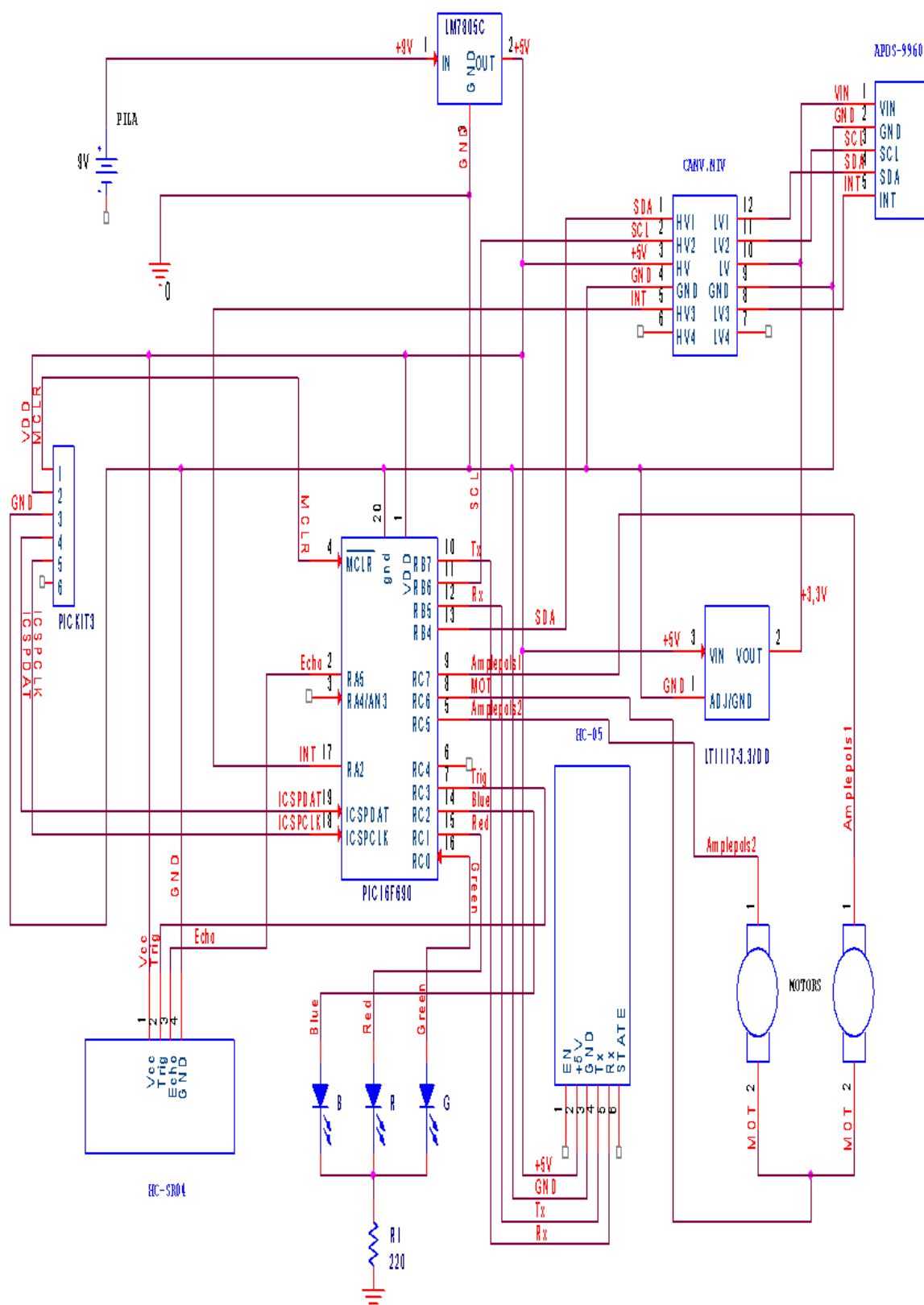


Figura 6.2- Esquema circuit electrònic amb OrCA

7. Mòdul Bluetooth

El dispositiu de bluetooth que s'ha connectat a la UART del microcontrolador per poder controlar el vehicle és el HC-05 (Figura 7.1). Aquest dispositiu permet connectar el microcontrolador i un telèfon mòbil mitjançant bluetooth **[10]**.

Aquest dispositiu té 4 pins per connectar: el pin 2 (VCC) a la tensió de 5 V, el pin 3 (GND) al terra, el pin 4 a la transmissió Tx del microcontrolador, que és el pin 10 o el pin 7 del port B, i el pin 5 a la recepció Rx del microcontrolador, que és el pin 12 o el pin 5 del port B. El pin 1, s'anomena Key, i s'utilitza per introduir comandes al dispositiu HC-05, no obstant, en aquest projecte el dispositiu només ha de rebre i transmetre dades i per tant, no s'ha utilitzat.

En aquest projecte, s'activarà el dispositiu HC-05 a través d'una aplicació de mòbil creada a partir de MIT App inventor **[9]**, una web que permet crear aplicacions Android.



Figura 7.1- Mòdul Bluetooth HC-05

8. HC-SR04

Al circuit del vehicle s'ha instal·lat un sensor d'ultrasons anomenat HC-SR04 per a que el vehicle s'aturi abans de contactar amb qualsevol objecte.

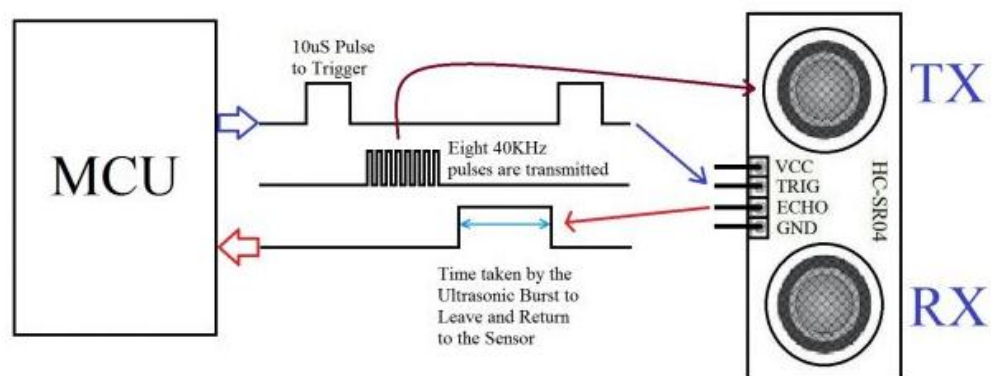
El sensor capta objectes que estiguin a una distància d'entre 2 cm i 400 cm, amb una exactitud de 3 mm, segons el datasheet del fabricant [12].

El sensor té 4 senyals:

- Senyal d'alimentació de 5V (VCC)
- Pin del Trigger (TRIG)
- Pin del Echo (ECHO).
- Terra (Gnd)

8.1 Funcionament

Primer de tot, com s'indica a la figura 8.1, el PIC 16F690 envia a la sortida del TRIG un pols de 10 μ s. Immediatament el sensor envia 8 ràfegues de 40 KHz per la transmissió TX que retornen per recepció RX. El microcontrolador rep un pols de 5 V per la senyal d'entrada ECHO. L'amplada del pols determina el temps que ha passat des de que les ones ultrasòniques han sortit de TX i han tornat a arribar a RX.



Working of HC-SR04 Ultrasonic Sensor

Figura 8.1- Esquema funcionament HC-SR04

Per calcular el temps que triga l'ona d'ultrasò en retornar al sensor, s'utilitza el registre TMR1 del Timer 1 que té 16 bits. Aquest registre és un comptador que va sumant per

cada període del rellotge intern. En el projecte, s'ha decidit fixar un valor inicial al TMR1 que determinés aturar el vehicle en detectar un obstacle a uns 6,8 cm.

Per calcular la distància mesurada pel sensor s'ha de seguir les següents equacions:

$$temps = TMR1 \cdot \frac{1}{\frac{f_{osc}}{4} \cdot \frac{1}{prescalador}} \quad (\text{Eq 6.1})$$

$$distància = \frac{(velocitatso \cdot temps)}{2} \quad (\text{Eq.6.2})$$

Primerament, el temps de retorn del senyal ECHO (Eq 6.1) s'ha de calcular tenint en compte la freqüència d'oscil·lació del microcontrolador. Segons el manual és de 4 MHz per defecte, però la freqüència important és l'efectiva del rellotge intern del PIC16F690 que és 4 vegades menor a aquesta, o sigui, de 1 MHz. Aquesta freqüència s'ha d'ajustar amb un prescalador que es selecciona mitjançant el registre T1CON i finalment el registre TMR1 pel període de la màquina

Un cop s'ha calculat el temps de retorn de l'ECHO, es pot calcular la distància entre el sensor i l'obstacle amb l'equació Eq 6.2. Per fer-ho només s'ha de pensar que les ones que emet l' ECHO són ones de so que es desplacen a la velocitat del so (velocitatso) a 34000 cm/s, multiplicar-ho pel temps calculat prèviament i dividir la distància entre 2, ja que la distància calculada seria la d'anada i la de tornada de l'ECHO.

Com s'ha explicat, s'ha de definir a part del registre TMR1, el registre T1CON. Aquest registre com es veu a la Figura 8.3 és necessari per configurar el Timer 1, ja que el posa en marxa i l'adapta a la necessitat del problema.

FIGURE 6-1: TIMER1 BLOCK DIAGRAM

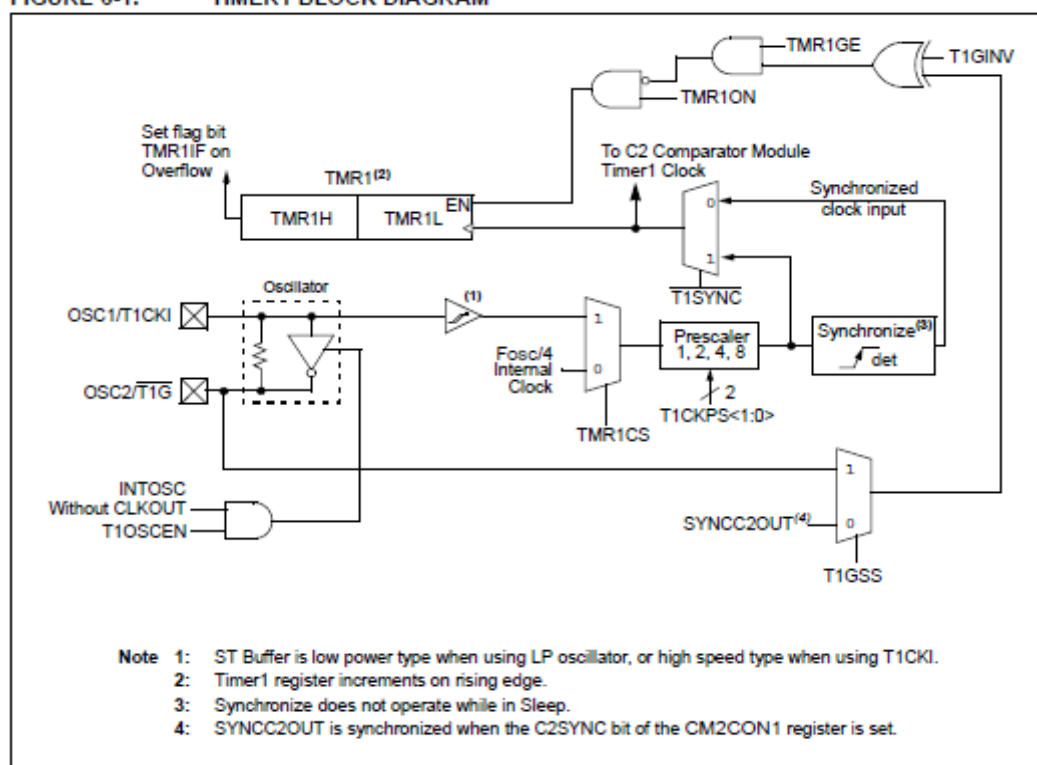


Figura 8.2- Diagrama de blocs del Timer 1

A la Figura 8.2 s'observa el detall del registre T1CON que s'ha explicat anteriorment per definir el temps, com per exemple, la freqüència interna del rellotge ($F_{osc}/4$), els prescaladors i el registre TMR1. En el projecte, el Registre T1CON es carrega amb el valor **00110001** perquè com s'observa a la taula cada bit activa alguna funció. Els bits 7 i 6 no s'utilitzen, els bits 5 i 4 activen el prescalador 1/8, el 3 desactiva el INTOSC, el 2 activa el Synchronized dock input, el 1 activa la freqüència rellotge intern ($f_{osc}/4$) i el bit 0 habilita el Timer 1.

REGISTER 6-1: T1CON: TIMER 1 CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
T1GINV ⁽¹⁾	TMR1GE ⁽²⁾	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
bit 7							bit 0

Figura 8.3- Registre T1CON

A la Figura 8.4 s'observa experimentalment un exemple de comportament del senyal ECHO com un pols. L'amplada del pols depèn de la distància entre el sensor i l'obstacle que capta, ja que com més gran sigui la distància, més gran serà l'amplada del pols, ja que mentre no reboti l'ona d'ultrasò amb cap obstacle, el senyal ECHO es mantindrà a nivell alt i igual a la tensió VCC.

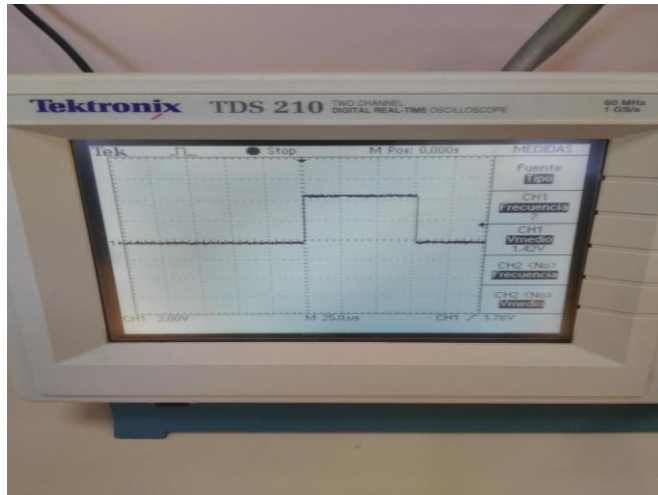


Figura 8.4- ECHO vist des de l'oscil·loscopi

9. Reguladors de tensió

En aquest projecte s'han utilitzat dos reguladors de tensions, ja que tots els components s'han d'alimentar a una tensió de 5 Volts, excepte el sensor de llum que s'ha d'alimentar a 3,3 V. Per alimentar el circuit s'ha utilitzat una pila de 9V.

9.1 Regulador 7805

El regulador 7805 disposa de tres pins de connexió: un per la tensió d'entrada, un altre per la tensió de sortida i un per connectar el dispositiu al terra. La disposició de cada pin ens la mostra el manual de funcionament del 7805 [7] com s'observa a la Figura 9.1.

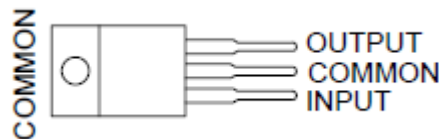


Figura 9.1- Imatge dels pins del regulador 7805

El regulador té una sèrie de condicions de funcionament:

- La tensió d'entrada haurà d'estar entre els valors de 7 V a 25 V.
- Intensitat màxima de 1,5 A.

La Figura 9.2 mostra la topologia del circuit amb el regulador per a transformar una tensió amb una altra fixa de 5 V.

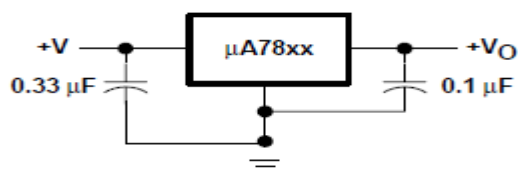


Figure 1. Fixed-Output Regulator

Figura 9.2- Esquema del circuit generador de voltatge xx amb el regulador 7805

9.2 Regulador LM1117

El regulador LM1117 disposa de tres pins per connectar la tensió d'entrada, la tensió de sortida i el terra comú. El pinout està indicat pel manual del dispositiu [5] a la Figura 9.3.

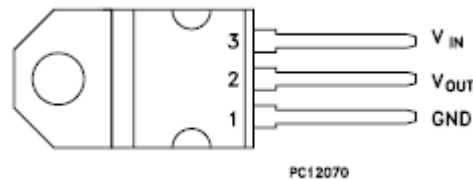


Figura 9.3- Imatge dels pins del regulador LM 1117.

El regulador té una sèrie de condicions de funcionament:

- La tensió d'entrada ha de ser de 4,75 V a 15 V.
- La intensitat de sortida és de 10 mA.

L'esquema del regulador per donar un corrent fixe de 3,3 V és el que es mostra al manual del LM1117 com s'observa a la Figura 9.4.

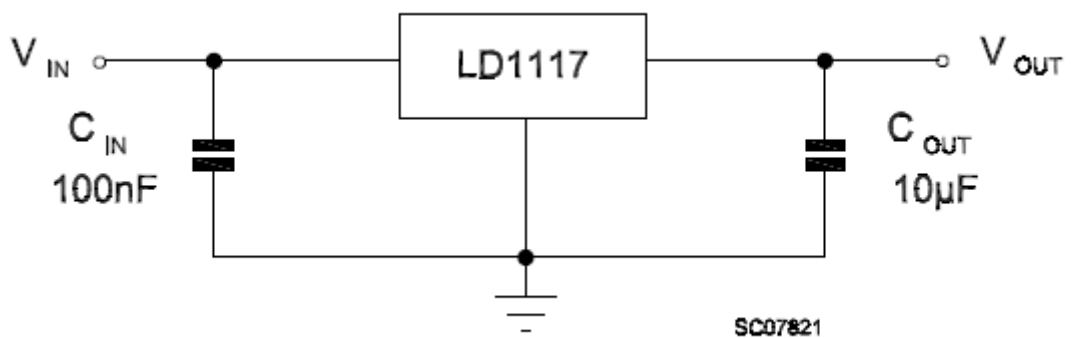


Figura 9.4- Esquema del circuit generador de voltatge 3,3V amb el regulador LM1117

10. Motor

10.1 Motor elèctric

El motor utilitzat és el TGP01D-A130-12215-48 (Figura 10.1), un motor DC d'imant permanent i escombretes de carbó amb una caixa reductora de 1:48. Aquest motor treballa en uns intervals de tensió de 3V a 9V i només pesa 35 g.

El fabricant [8] ofereix una taula (Taula 10.1) amb els paràmetres més importants dels motors:

MODEL		VOLTAGE		NO LOAD		AT MAXIMUM EFFICIENCY					STALL		
		OPERATING RANGE	NOMINAL	SPEED	CURRENT	SPEED	CURRENT	TORQUE		OUTPUT	TORQUE		CURRENT
			V	r/min	A	r/min	A	N·m	kgf·cm	W	N·m	kgf·cm	A
TGP01D-A130	10300-288	3.0 - 12.0	6	21	0.10	19	0.17	0.05	0.47	0.18	0.18	1.8	0.6
	14175-120	3.0 - 6.0	4.5	90	0.20	72	0.50	0.03	0.3	0.38	0.14	1.4	1.25
	12215-48	3.0 - 9.0	3.0	110	0.12	85	0.20	0.01	0.11	0.13	0.027	0.27	0.45
	18100-220	3.0 - 6.0	3.0	50	0.25	34	0.45	0.07	0.71	0.52	0.24	2.4	1.1

Taula 10.1 – Taula dels paràmetres d'un motor elèctric

Els paràmetres més importants són la velocitat sense càrrega de 110 rpm, el corrent bloquejat de 450 mA, el corrent sense càrrega (règim permanent) de 120 mA i un parell motor bloquejat de 0,27 N·m.

Per poder fer funcionar els dos motors es necessita un amplificador de corrent per poder alimentar adequadament els motors a partir del PIC 16F690 (ja que aquest només pot lliurar 25mA, insuficients pel funcionament correcte del motor). En aquest projecte s'utilitzarà el L293D.



Figura 10.1- Motor elèctric TFP01-A130

10.2 PWM programat

Els motors funcionen a partir d'un senyal PWM (pulse width modulation), o sigui, un tren de polsos a 5 V dels quals se'n pot modificar la seva amplitud d'ona per poder variar la seva velocitat. Al principi, es va plantejar utilitzar el mode PWM del PIC 16F690 per un motor i crear una funció software que generés la mateixa forma d'ona per l'altre motor, però es produïen molts problemes per controlar cada roda i al final es va decidir crear una única funció que creés un tren de polsos per a cada motor.

Per generar el senyal de polsos s'ha realitzat la funció de la Figura 10.2:

```

if (INTCONbits.T0IF)    // Interrupció del TMR0 cada 93 us
{
    // S 'incrementa compte

    if (compte == amplepols1) //Quan l'incrementador compte sigui igual a amplepols + 1
    {
        P0L1 = 0; // El senyal del motor passa a ser 0
    }
    if (compte == amplepols2) //Quan l'incrementador compte sigui igual a amplepols + 1
    {
        P0L2 = 0; // El senyal del motor passa a ser 0
    }
    compte++;
    if (compte==100)    //quan arriba al final del període, inicialitzem el PWM
    {
        compte=0;
        P0L1=1;
        P0L2=1;
    }

    TMR0= 240;
}

```

Figura 10.2- Funció del motor del programa

El que s'ha fet és utilitzar el Timer 0 per poder controlar el període de cada pols. Aquest timer, com s'ha vist als tipus a l'apartat **Interrupcions**, genera una interrupció per desbordament.

Si s'agafa un valor de TMR0 (precàrrega) de 240, el Timer 0 fa 15 cicles de 4µs fins desbordar-se i, per tant, arriba la interrupció del Timer 0 cada 60 µs. Ara bé, la interrupció necessita un temps per executar-se que s'haurà de sumar al temps anterior per saber quin serà el període de cada pols.

En aquest cas el període és de 93 μ s. Aquest temps és variable i depèn del valor del Timer 0 ja que com més petit sigui el seu valor, més gran serà el període i per tant, més petita serà la freqüència dels polsos. El fet que el període sigui de 93 μ s és degut que s'ha realitzat un càlcul per aproximar la freqüència dels polsos a 100 Hz, en aquest cas 107,53 Hz.

Com s'observa a la Figura 10.2, s'ha creat un comptador que vagi de 0 a 100 per a dividir el període de cada pols en 100 parts. Això permet saber quan el valor del pols canvia de 1 a 0, que coincidirà amb l'amplitud de cada pols.

A l'apartat **Aplicació Mòbil** s'explicarà com a partir d'una dada enviada via Bluetooth es modifica l'amplitud del pols.

L'altra qüestió important en el moviment dels motors és com commuten entre girar endavant a endarrere. Per realitzar aquest canvi s'ha creat una variable anomenada MOT que adquireix el valor de 0 ó 1. El que fa aquest canvi és invertir la polaritat del motor, ja que si el MOT és 0, la tensió que dóna el PWM serà més gran o igual i per tant el motor gira cap endavant. En canvi, si el MOT és 1, la tensió que dóna el PWM és més petita o igual i per tant, el sentit de gir és cap a l'altre sentit.

10.3 PWM Pulse Width Modulation

Al principi del projecte es va treballar amb el mode PWM (Pulse Width Module) del mòdul SSP del microcontrolador per generar els trens de polsos per als dos motors, però al decidir fer girar el vehicle i controlar només un dels motors amb el PWM i l'altre motor amb un altre senyal es van tenir problemes al controlar el mòdul i es va optar per la solució explicada a l'apartat anterior. No obstant, s'explicarà com fer servir aquest mòdul perquè és una aplicació molt interessant que ens mostra el microcontrolador per emetre un senyal amb forma de pols sense crear-lo per software al programa.

Per utilitzar aquest mòdul, segons el manual del PIC 16F690, s'han de connectar els motors al senyal CCP1CON que és accessible al pin RC5 del microcontrolador.

Per activar el senyal PWM s'han d'activar els següents senyals:

- CCP1, activa el senyal PWM
- els bits 4 i 5 del senyal CCP1CON (CCP1L)
- PR2, activa el Timer 2 i indica la seva freqüència: en el projecte és de 250 Hz.

A part d'aquests senyals, s'hauran de definir altres variables per poder controlar l'amplada del pols i poder ajustar la freqüència fixada de 250 Hz al període de màquina del microcontrolador de 1 us.

Per poder controlar l'amplada del pols, el manual del PIC 16F690 ens dóna la informació de la Figura 10.3:

$$\text{Pulse Width} = (\text{CCP1L:CCP1CON} \langle 5:4 \rangle) \cdot T_{osc} \cdot (\text{TMR2 Prescale Value})$$

Figura 10.3- Fórmula PWM de l'amplada de pols

Per poder ajustar la freqüència del PWM també ens dóna l'equació mostrada a la Figura 10.4:

$$PWM\ Period = [(PR2) + 1] \cdot 4 \cdot T_{osc} \cdot (TMR2\ Prescale\ Value)$$

Note: $T_{osc} = 1/F_{osc}$

Figura 10.4- Fórmula període del PWM

Per poder ajustar la freqüència es necessitarà un valor de preescala que en aquest cas serà de 1/16, ja que per treballar el PWM al mateix període de màquina del microcontrolador, que és de 1 us, s'haurà de rebaixar el període de 4 ms (freqüència de 250 Hz) que es multiplica per 4 segons la fórmula extreta del manual.

El CCP1CON ha de tenir els bits 4 i 5 amb un 1 perquè és la forma que s'activa el mode del PIC16F690.

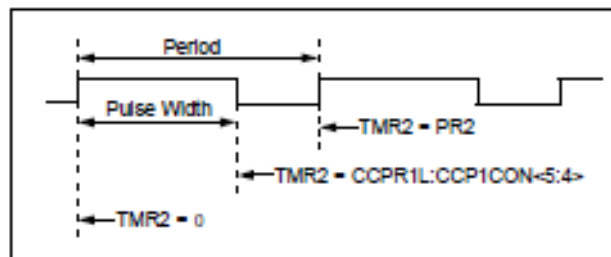


Figura 10.5- Mostra del valor TMR2

La Figura 10.5 mostra com influeix el Timer 2 a l'hora de crear el microcontrolador un pols. El timer, amb la variable TMR2, defineix l'amplitud del pols i el període de cada pols amb el senyal PR2, que guarda la freqüència.

Per tal que surti un pols del microcontrolador, és important definir el pin RC5, que conté el senyal CCP1, com a sortida.

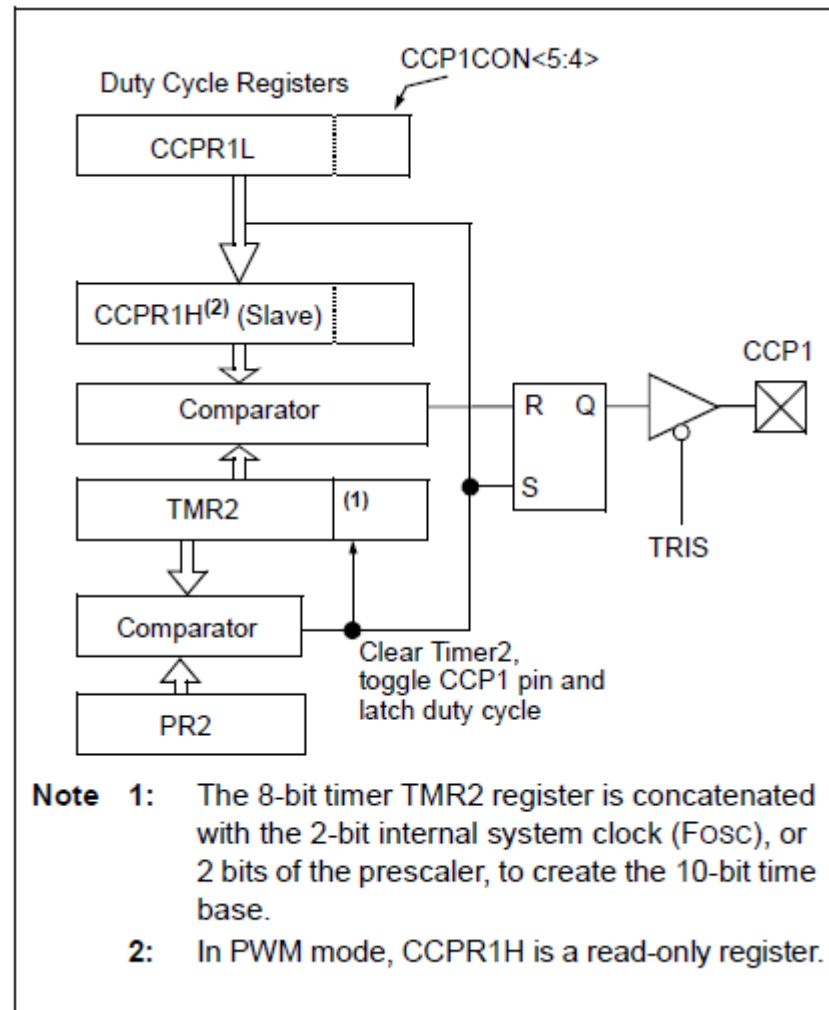


Figura 10.6- Diagrama de blocs del PWM

El manual del PIC16F690, a part de donar les fórmules de càlcul de totes les variables del PWM, també ens mostra el diagrama de blocs (Figura 10.6) que mostra com s'ha dit que els senyals PR2 i TMR2 s'encarreguen de fixar el període del perifèric i com els senyals CCPR1 i TMR2 defineixen que el perifèric serà un PWM amb una determinada longitud de pols. Per últim, aquest senyal s'haurà de definir com a sortida amb un 0 al TRIS del pin RC5 (CCP1).

11. Amplificador de corrent L293D

En aquest projecte sorgeix una problemàtica bastant interessant entre la potència que dóna la sortida del microcontrolador i la potència que necessiten els motors per funcionar. Com s'ha explicat, el PIC16F690 proporciona una tensió en forma de polsos que anirà variant la seva amplitud per tal de variar la velocitat. Ara bé, el microcontrolador té un corrent de sortida molt petit de un màxim de $67\ \mu\text{A}$ per a cada sortida del microcontrolador cap a cada motor.

Per tal de solucionar aquest problema, s'ha hagut d'afegir un amplificador de corrent, exactament un Quàdruple High-current half H-driver, anomenat L293D [6].

Aquest amplificador necessita dues alimentacions com es veu a la Figura 11.1, ja que està basat en un circuit lògic que estarà alimentat per una tensió V_{CC1} i un circuit que estarà connectat a l'element que es vol alimentar, en aquest cas, cap als dos motors.

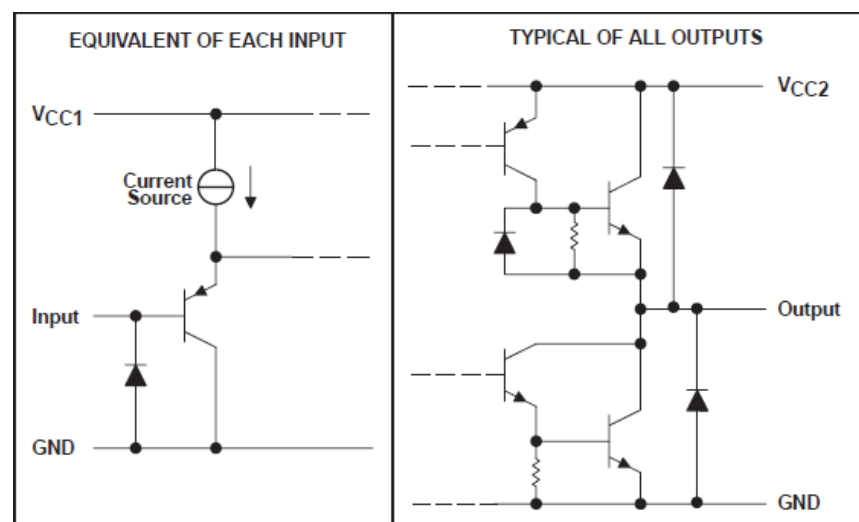


Figura 11.1- Circuit lògic i circuit de l'amplificació d'un motor

La V_{CC1} pot prendre valors de 4,5 V a 7 V, mentre que la V_{CC2} pot prendre des de la V_{CC1} fins a 36V. A més pot amplificar la intensitat d'entrada fins a 600 mA.

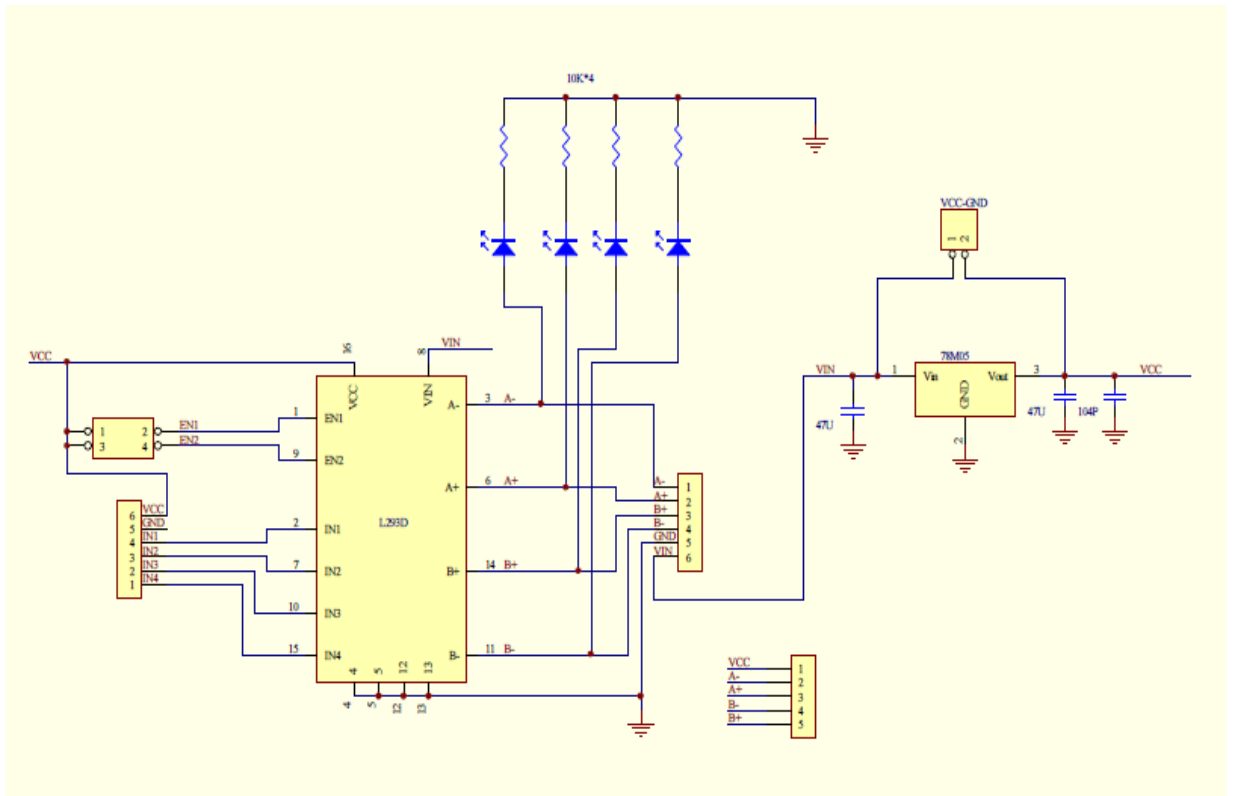


Figura 11.2- Esquema del circuit utilitzat amb l'amplificador L293D

La figura 11.2 mostra tot l'esquema del circuit electrònic que envolta el L293D [11].

S'observa com la VCC del pin 16 és l'alimentació de la lògica i l'alimentació del pin 8 VIN és la que alimenta la part del motor.

Les entrades IN són les d'entrada i les A i les B són les sortides de l'amplificador que aniran als motors. Els leds que es mostren a la imatge són els que ens indiquen si passa tensió i són bastant útils per comprovar d'una manera visual la tensió que es dona als motors ja que les seves lluminositats varien.

La part més a l'esquerra ens mostra un regulador de tensió a 5 V per permetre ficar tensions d'entrada de fins a 36 V.

Afegint aquest element, s'ha aconseguit donar un corrent de 85 mA als motors per al seu funcionament.

12. Canviador de nivell lògic

Per ajustar els mateixos nivells digitals entre el microcontrolador PIC 16F690 i el sensor APDS-9960 és necessari la utilització d'un canviador de nivell. És evident que sense aquest dispositiu no hi haurà comunicació entre els dos elements, ja que un '1' són 5V al microcontrolador i són 3,3 V al sensor de llum.

Per tant, s'ha identificat com a tensió alta els 5 V i com a tensió baixa els 3,3 V. Tot seguit, s'ha col·locat els 3 senyals (SDA, SDA i INT (RA2)) a aquest dispositiu per a igualar els senyals digitals, és a dir, que un '1' del microcontrolador quan passi pel canviador també ho sigui del sensor.

La Figura 12.1 mostra l'interior del canviador que té un mosfet BSS 138 per a cada parella d'entrada i sortida.

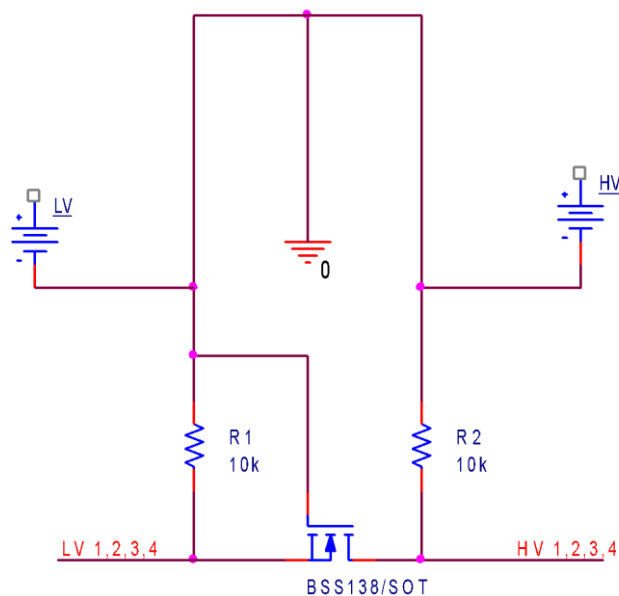


Figura 12.1- Esquema del canviador de nivell per Orcad

13. Sensor de llum

El sensor de llum utilitzat per controlar el vehicle ha estat el APDS-9960 [1]. Aquest sensor té molta versatilitat ja que té vàries aplicacions:

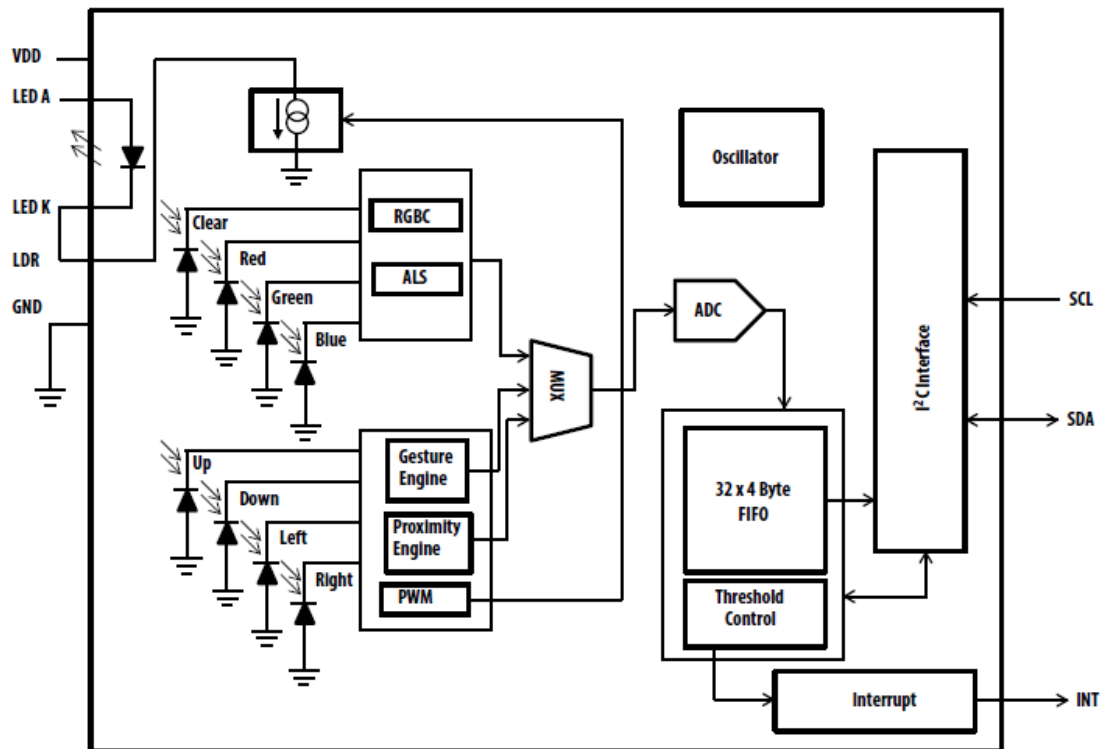
- Detector de gestos
- Captador de colors
- Captador de llum ambiental
- Desactivador de la pantalla tàctil d'un mòbil
- Canviador de commutació mecànic

Tot i això, les seves característiques principals es resumeixen a ser un detector de llum ambiental i colors en format RGB, un detector de proximitat i un detector de gestos. En aquest projecte només s'utilitzarà el detector de llum ambiental i colors en format RGB per tal de simplificar les ordres.

Per tant, només es treballarà en el bloc de Color i ALS que ens permet adquirir informació de la llum que capta el sensor, en aquest cas, només del color de la llum en format RGB i de la seva intensitat lluminosa.

Com s'observa a la Figura 13.1, la part del RGBC i ALS hi ha un conjunt de fotodíodes que són capaços de descompondre la llum en vermell (Red), verd (Green), blau (Blue) i llum clara (Clear). Dins d'aquest bloc hi ha una sèrie de bloquejadors que filtren els rajos UV (ultraviolats) i IR (infrarojos). Un cop s'ha descomposat la llum, aquesta informació passa a un bloc que la uneix amb la informació de les dues altres parts funcionals, el captador de gestos i el captador de proximitat. Tota informació captada pel sensor es transforma a digital a partir del convertidor ADC i s'emmagatzema al registre corresponent dins la FIFO a l'espera de rebre ordres de lectura del PIC 16F690 a través del bus I2C.

Functional Block Diagram

**Figura 13.1- Diagrama de blocs del sensor APDS-9960**

13.1 Condicions de Funcionament

Per a funcionar el sensor s'ha de seguir una sèrie d'especificacions. El manual ens mostra unes taules on hi apareixen els paràmetres més significatius de funcionament i els rangs de valors de les senyals SCL (signal clock) i SDA (signal data):

- Tensió d'alimentació (VDD) haurà d'estar entre 2,4 V i 3,6 V. En el projecte s'alimentarà a 3,3 V.
- El rang de temperatures on el sensor funciona adequadament és entre -30 °C i 85°C.
- El corrent subministrat (IDD) aplicant el detector de colors serà de 790 µA.
- Els senyals SCL i SDA donen una tensió d'entre 1,26 V i la VDD, que en aquest cas serà de 3,3 V.

13.2 I2C

El bus I2C és un bus de comunicació en sèrie i síncrona creat per Philips al 1980 que es basa en establir una relació de mestre-esclau on el mestre serà el PIC16F690 i l'esclau el sensor APDS-9960. Actualment és un bus molt utilitzat per treballar amb microcontroladors i els seus perifèrics. La seva principal característica és que té dues línies per transmetre informació: una línia per la senyal de rellotge (SCL) i una altra per la senyal de dades (SDA), aquesta última bidireccional ja que el mestre pot escriure o llegir a l'esclau.

13.2.1 Funcionament

El bus I2C proporciona tres tipus de transacció com es veu a la Figura 13.2: lectura, escriptura i una combinada.

Si es vol realitzar una escriptura, el mestre ha de dir l'adreça de l'esclau, l'operació Write amb un 0, el registre on es vol escriure i la dada. Si es vol realitzar una lectura, el mestre ha d'indicar l'adreça de l'esclau, l'operació de Read amb un 1 i que l'esclau envii la dada al mestre.

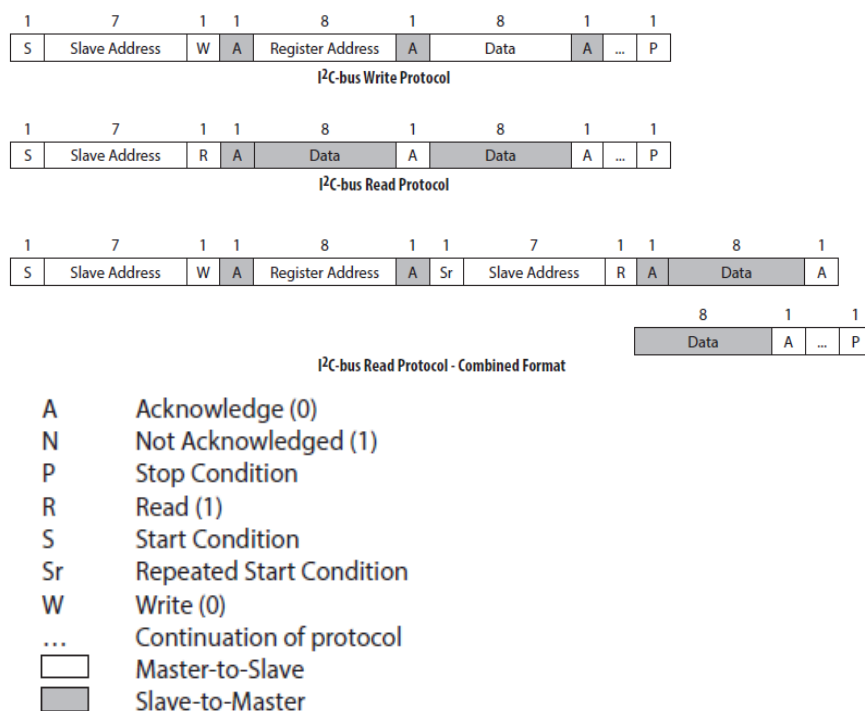


Figura 13.2- Protocol I2C

La transacció combinada és realitzar les dues operacions a la vegada, ja que el bus és bidireccional.

Per tal d'establir una correcta comunicació entre el mestre i l'esclau, aquest darrer ha d'enviar un senyal d'un bit anomenat Acknowledge (ACK) que confirma la connexió entre els dos elements. Com s'ha explicat a l'apartat d'I2C del microcontrolador, els senyals SDA i SCL estan connectats a alta impedància quan no hi ha connexió, és a dir, que hi arriba un '1'. Per tant, per canviar l'estat a connectat s'ha d'enviar un senyal de ACK de '0' (Figura 13.3).

Aquest senyal sempre apareix al bit número 9 del rellotge, ja que els anteriors bits serien els de l'adreça i la ordre o els d'una dada. Això significa que aquest novè bit d'aquest senyal fa la funció de confirmar la connexió entre el microcontrolador i el sensor. Quan s'acaba la connexió, ACK torna a enviar un valor de '1' per tornar a l'estat de no connexió (Figura 13.4).

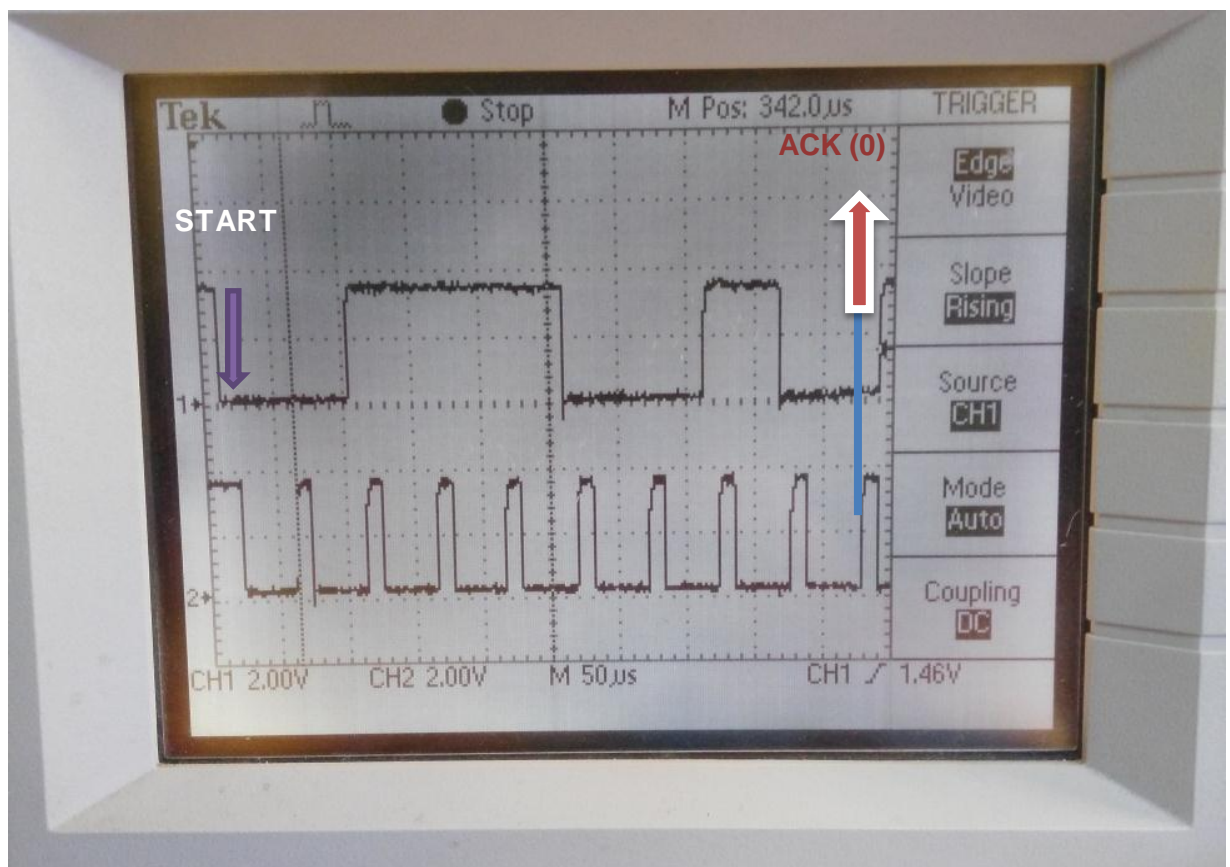


Figura 13.3- Senyals SDA i SCL amb connexió

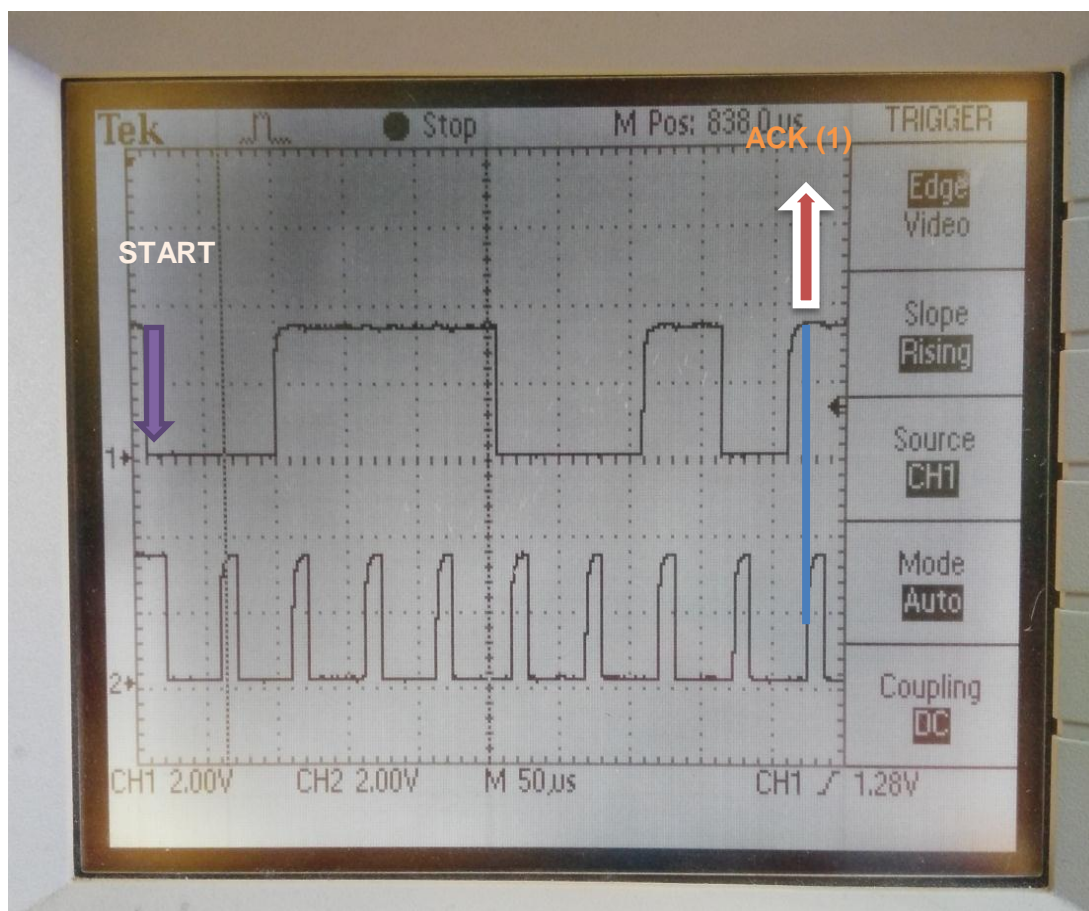


Figura 13.4- Senyals SDA i SCL amb desconnexió

En el programa fet, s'ha adjuntat el fitxer APDS-9960 (Figura 13.5) on es pot observar com es segueix la dinàmica explicada amb les funcions `wireWriteDataByte` i `wireReadDataByte`.

```

/*Writes a single byte to the I2C device and specified register*/
int wireWriteDataByte(unsigned char reg, unsigned char val)
{
    i2c_start();
    i2c_out_byte(APDS9960_I2C_ADDR_W);
    i2c_ack();
    i2c_out_byte(reg | 0x80);
    i2c_ack();
    i2c_out_byte(val);
    i2c_ack();
    i2c_stop();

    return 1;
}

/*Reads a single byte from the I2C device and specified register*/
unsigned char wireReadDataByte(unsigned char reg)
{
    /* Indicate which register we want to read from */
    char val = 0;
    i2c_start();
    i2c_out_byte(APDS9960_I2C_ADDR_W);
    i2c_ack();
    i2c_out_byte(reg | 0x80);
    i2c_ack();
    i2c_start();
    i2c_out_byte(APDS9960_I2C_ADDR_R);
    i2c_ack();
    val = i2c_in_byte();
    i2c_nack();
    i2c_stop();
    return (val);
}

```

Figura 13.5- Funcions d'escriure i llegir de la llibreria I2C

Finalment, només indicar que en un bus I2C els senyals de SCL i SDA estan en senyal lògic alt (alta impedància) a partir d'una resistència de pull-up com ens indica el manual a la Figura 13.6.

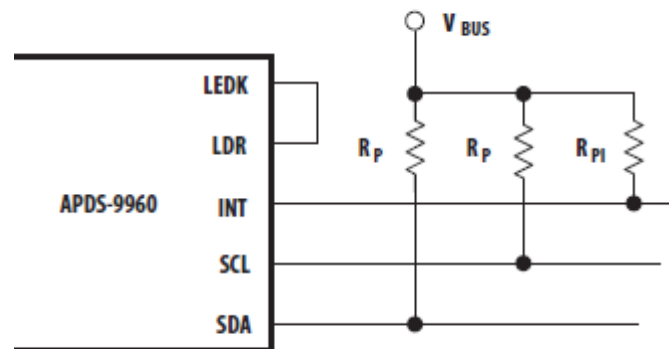


Figura 13.6- Pull-up senyals SDA, SCL i INT

13.3 Interrupció INT

El sensor de color informa de que disposa d'un nou resultat de conversió de color mitjançant l'enviament d'un senyal d'interruptió externa cap al microcontrolador. És llavors quan el PIC pot llegir-ne els valors i, un cop feta aquesta lectura, el sensor APDS-9960 (slave) ha de rebre l'activació d'una nova conversió. Per fer-ho s'ha de netejar l'adreça amb la comanda PICLEAR.

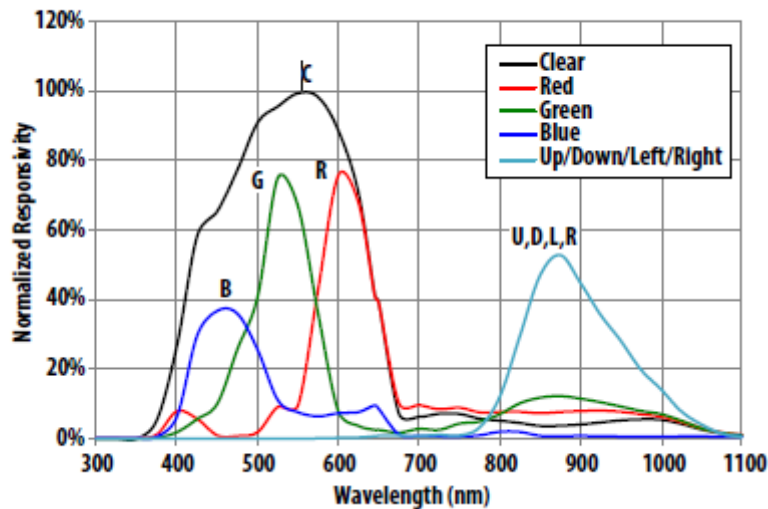
Aquesta interrupció externa és el senyal INT i només s'activa quan el sensor capta una variació de la llum que rep. En aquest projecte, s'ha utilitzat la interrupció externa que pot realitzar el microcontrolador al pin RA2 (INT).

13.4 RGB

La part del sensor que rastreja el color ho fa seguint el codi de classificació de colors RGB (Red, Green, Blue), que separa qualsevol color en aquests tres components. Cada raig de llum que es captat pel sensor, passa un filtratge dels rajos UV i dels rajos IR quedant només l'espectre visible.

Quan el sensor ha rastrejat la llum que ha captat i només queda la part visible de la llum, és a dir l'espectre de color entre el vermell i el blau, divideix la informació del color segons

la longitud d'ona de la llum, ja que la diferència de color es relaciona amb la longitud d'ona.



Gràfic 13.7- Gràfic de la resposta del sensor segons la longitud d'ona

El manual ens mostra el Gràfic 13.7 com respon davant la llum i la separa segons la seva longitud d'ona. Per exemple, si rastregem una llum amb component verda d'una longitud d'ona de 520nm, el sensor donarà un component de verd molt alt respecte als altres dos colors [14]. En canvi si es fa l'anàlisi invers que en lloc de posar una ona d'una longitud d'ona determinada [2], es posa una llum vermella davant del sensor, segons el gràfic es pot deduir que la longitud d'ona de la llum que capa serà d'uns 600 nm.

La diferència del codi RGB amb les dades de cada color és que el sensor no donarà un valor màxim absolut de vermell, de verd i de blau, ja que els valors els compara amb la intensitat lluminosa que rep el sensor de la llum com es veu en el gràfic.

No obstant, el manual del sensor ens mostra la Figura 13.8 que és una taula amb tres longituds d'ona i els % de rati dels 3 colors segons la intensitat lluminosa (Clear) amb les condicions d'alimentació de 3V i temperatura de treball de 25°C. Com s'observa traiem la mateixa informació que al gràfic, ja que a la llum amb una longitud d'ona a 465 nm el color hi predomina el blau, a 525 nm hi predomina el verd i a 625 nm hi predomina el vermell.

Optical Characteristics, $V_{DD} = 3\text{ V}$, $T_A = 25\text{ }^{\circ}\text{C}$, $\text{AGAIN} = 16\times$, $\text{AEN} = 1$ (unless otherwise noted)

Parameter	Red Channel		Green Channel		Blue Channel		Units	Test Conditions
	Min	Max	Min	Max	Min	Max		
Irradiance responsivity ^[1]	0	15	10	42	57	100	%	$\lambda_D = 465\text{ nm}$ ^[2]
	4	25	54	85	10	45		$\lambda_D = 525\text{ nm}$ ^[3]
	64	120	0	14	3	29		$\lambda_D = 625\text{ nm}$ ^[4]

Figura 13.8- Resposta segons la longitud d'ona

Un cop el sensor de llum ha captat el color de la llum en RGB i la seva intensitat lluminosa, la guarda a uns registres anomenats RGBCDATA. Es recorda que seran 8 registres, un byte MSB i un byte LSB per a cada color RGB i per la intensitat lluminosa (CLEAR).

En el cas del sensor de llum del projecte, les dades seran només de lectura (Read) i seran les adreces que es mostren a la Figura 13.9:

Field	Address	Bits	Description
CDATAL	0x94	7:0	Low Byte of clear channel data.
CDATAH	0x95	7:0	High Byte of clear channel data.
RDATAL	0x96	7:0	Low Byte of red channel data.
RDATAH	0x97	7:0	High Byte of red channel data.
GDATAL	0x98	7:0	Low Byte of green channel data.
GDATAH	0x99	7:0	High Byte of green channel data.
BDATAL	0x9A	7:0	Low Byte of blue channel data.
BDATAH	0x9B	7:0	High Byte of blue channel data.

Figura 13.9- Adreces de cada color

Com s'observa, cada valor de component RGB serà representat per dos bytes, un de MSB i un altre de LSB, sent el byte de menys pes el que serà llegit primerament.

13.5 Codi RGB

El fet de descomposar un color en format RGB és molt útil a l'hora d'identificar colors, ja que és molt útil guardar un color segons la combinació de Vermell, Verd i Blau com si cada color fos únic en l'espai de colors. Normalment, els colors es guarden en 3 variables d'un byte perquè es considera que és suficient tenir informació de 2^4 colors.

Ara bé, el sensor APDS-9960 és capaç d'emmagatzemar 2 bytes per color, i per tant, de donar més precisió sobre la tonalitat de cada color.

Gràcies a la codificació RGB es pot classificar qualsevol color amb només tres paràmetres. Això permet representar tots els colors en forma de cub (Figura 13.10) en un pla 3D amb els colors vermell, verd i blau com a eixos i extreure una informació interessant sobre la formació de colors:

- La unió de 2 dels 3 colors RGB formen els colors primaris (magenta, cyan, groc)
- La llum blanca emet tots els colors, per tant al vèrtex (1,1,1) el color serà el blanc.
- El negre és l'absència de llum, per tant quan no hi hagi llum els valors

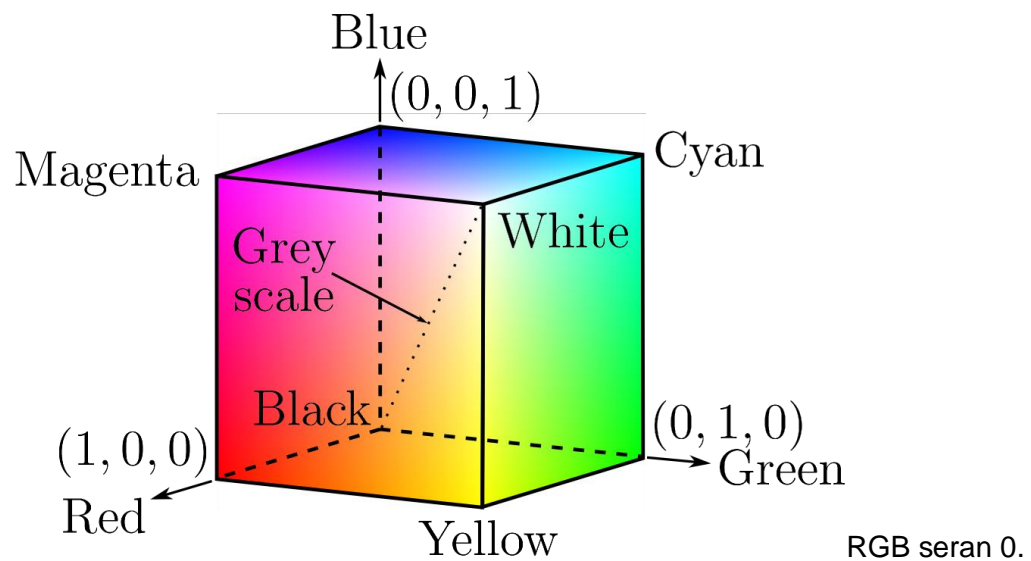


Figura 13.10- Cub RGB

14. Aplicació Mòbil

14.1 Introducció a l'aplicació

L'aplicació de mòbil s'ha realitzat a través de MIT app Inventor, una web que permet crear aplicacions per mòbils sense haver de programar i només interactuant visualment uns blocs. Aquesta web neix d'un projecte entre l'empresa Google i la Massachusetts Institute of Technology (MIT) l'any 2009 per a que la gent consumidora de tecnologia tingui la oportunitat de fabricar-se les seves pròpies aplicacions sense tenir un coneixement gaire ampli sobre software.

Per crear l'aplicació, l'usuari s'ha de fixar en la llista d'elements que apareixen a l'esquerra de la pàgina web (Figura 14.1) amb que primer de tot s'ha de dissenyar l'aplicació, és a dir, s'ha d'afegir els elements de la interfície del dissenyador (Designer) a una pantalla petita que posa Viewer.

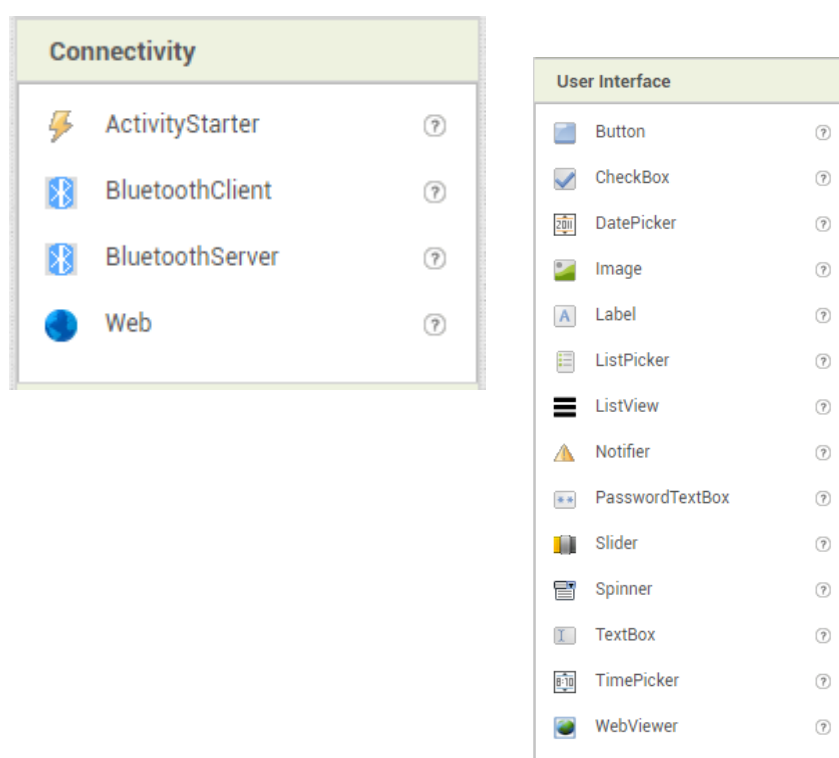


Figura 14.1- Connectivitat Bluetooth i interfície

Com s'observa a la Figura 14.2, el programa permet manipular fàcilment botons, quadres de texts, etc. També permet utilitzar rellotges que contenen cada X ms i realitzar una connexió amb un dispositiu via Bluetooth, que en aquest projecte serà el HC-05.

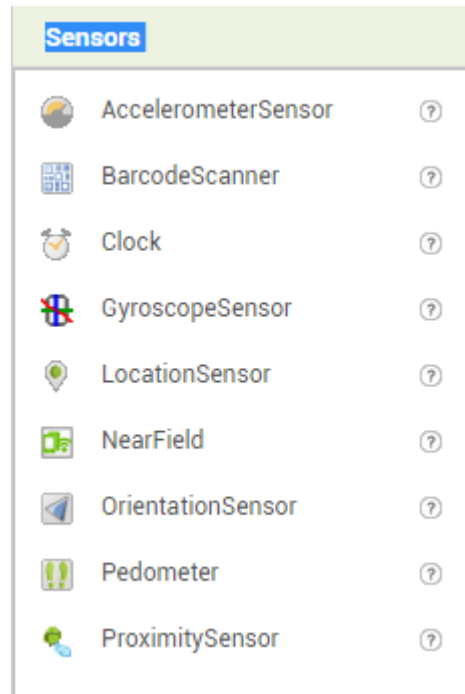


Figura 14.2- Sensors de APP-Inventor

Un cop es tenen els elements seleccionats, l'usuari a la interfície de blocs, ha de lligar cada element a partir d'una sèrie de blocs funcionals. L'aplicació permet utilitzar una sèrie de blocs predeterminats més els blocs que aporta cada element seleccionat al dissenyador com s'observa a la Figura 14.3.

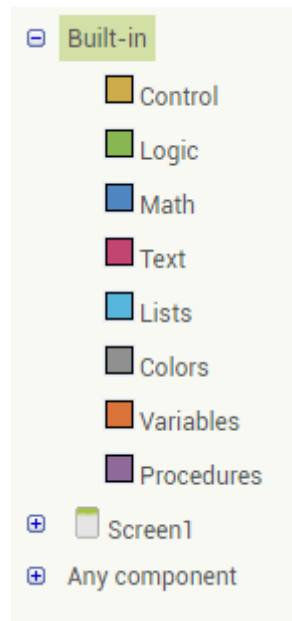


Figura 14.3- Funcions dins dels blocs a APP-Inventor

9.1. Aplicació de mòbil del projecte

L'aplicació consta de dues parts diferenciades, ja que una part controlarà el vehicle i l'altra ens mostrarà les dades que capta el sensor de llum.

A la Figura 14.4 es mostra el que sortirà a la pantalla del mòbil. S'observa com hi ha un botó superior, el Text for ListPicker1, que s'encarregarà de connectar-se via bluetooth amb el mòdul HC-05. A continuació, es veuen dos Sliders i 5 botons (GO, STOP, Endarrere, LEFT i RIGHT) dins d'una capseta que s'encarregaran de controlar el vehicle i finalment s'observa una taula on es mostraran els valors captats pel sensor de llum.

A la part de sota, es veuen aquells elements que també seran utilitzats, però que no sortiran a la pantalla del telèfon mòbil com són la connexió Bluetooth i un rellotge.



Figura 14.4 Viewer de l'aplicació

14.2 Connectivitat Bluetooth

El primer pas per a que l'aplicació funcioni és que el telèfon mòbil es pugui connectar amb el dispositiu de bluetooth HC-05. Per això es crea un botó anomenat ListPicker 1 i se'l programa amb el bloc de la Figura 14.5.



Figura 14.5- Funció ListPicker

La funció BeforePicking permet realitzar una operació sense haver activat cap element prèviament. D'aquesta forma podem connectar tots els elements a la connexió del bluetooth gràcies a l'element que surt a la imatge del Viewer BluetoothClient1 i no només això, sinó que cridant-ho amb la instrucció AddressesAndNames el mòbil es pot connectar amb qualsevol dispositiu que tingui aparellat via bluetooth.

Un cop es connecta s'observa que el led del dispositiu HC-05 redueix la freqüència d'intermitència indicant que la connexió ja està establerta.

14.3 Control vehicle

El control del vehicle s'ha dut a terme amb dos Sliders i 5 botons. Els Sliders serveixen per enviar una dada i que aquesta estigui dins d'un interval definit, en canvi, els botons només enviaran una dada que contingui un valor predeterminat.

Com s'ha explicat en la secció de la UART, el microcontrolador pot transmetre i rebre dades de la mida d'un byte, és a dir, valors de 0 a 255 i per tant no es pot enviar dades més grans de 255.

14.3.1 Sliders

Per l'Slider 1 (groc), s'ha creat a la part de blocs l'interval de dades que envia al microcontrolador [0,100]. Aquest interval s'ha dividit en dos parts dividides pel valor 50 on la part de l'esquerra s'encarregarà que la velocitat del vehicle vagi cap endarrere i la part de la dreta serà per quan la velocitat del cotxe vagi cap endavant. Per tant, quan se li

enviï el byte 50, la velocitat i la tensió seran 0 i quan se li enviï els bytes 0 i 100 la velocitat i la tensió seran màximes (encara que cadascuna en un sentit diferent).

Per l'Slider 2 (vermell), s'ha definit que enviï dades de l'interval de [101,201]. L'estructura de l'slider és la mateixa als dos, però ara el byte que partirà l'slider és el 150. A partir d'aquest valor, els valors compresos entre 101 i 149 afegeixen un increment de tensió i per tant, de velocitat, a la roda de la dreta per a que el vehicle giri a l'esquerra. En canvi, els valors compresos entre 151 i 201 fan que la roda de l'esquerra giri a més velocitat que la dreta i que per tant el vehicle giri cap a la dreta.

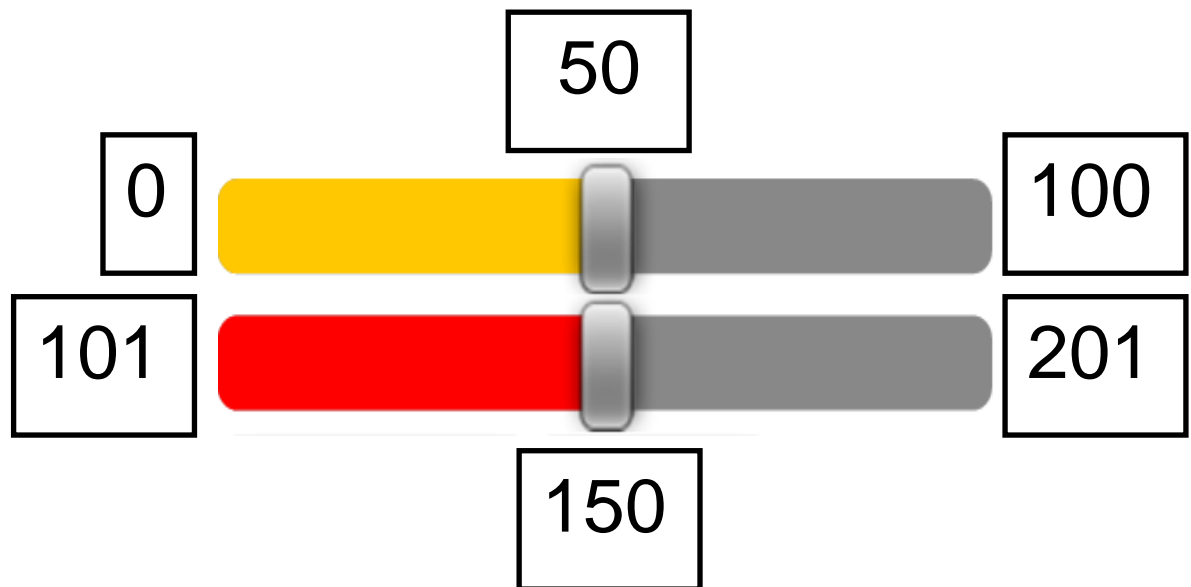


Figura 14.6- Imatge dels dos Sliders amb els seus bytes més importants

Aquesta forma de fer girar un vehicle es basa en el concepte mecànic del CIR (Centre Instantani de Rotació), ja que un vehicle a diferents velocitats a les rodes té un CIR definit i per tant gira i un vehicle amb les dues velocitats iguals no el té definit i per tant només es trasllada.

És important destacar que l'Slider 1 el que fa és enviar un byte que defineix l'ample dels polsos que alimenten els motors segons es mostra al programa. La "DADA" és el valor enviat a través de l'aplicació com s'observa a la Figura 14.7, i és la que determina que el

vehicle es trasllada si el valor és més petit que 101. A més, s'observa que el càlcul de la velocitat velRecta és simètrica a 50.

```

if (DADA < 101)
{
    if (DADA >= 50)
    {
        MOT=0;
        velRecta=(DADA-50)*2;
        amplepols2 = velRecta;
        amplepols1 = velRecta;
    }
    else
    {
        MOT=1;
        velRecta=100-(50-DADA)*2;
        amplepols2 = velRecta;
        amplepols1 = velRecta;
    }
}

```

Figura 14.7- Condició d'anar recte del programa

Si observem la Figura 14.8, s'aprecia com l'enviament de dades inferiors a 101, és a dir, per l'Slider 1 fa variar l'amplitud dels polsos.

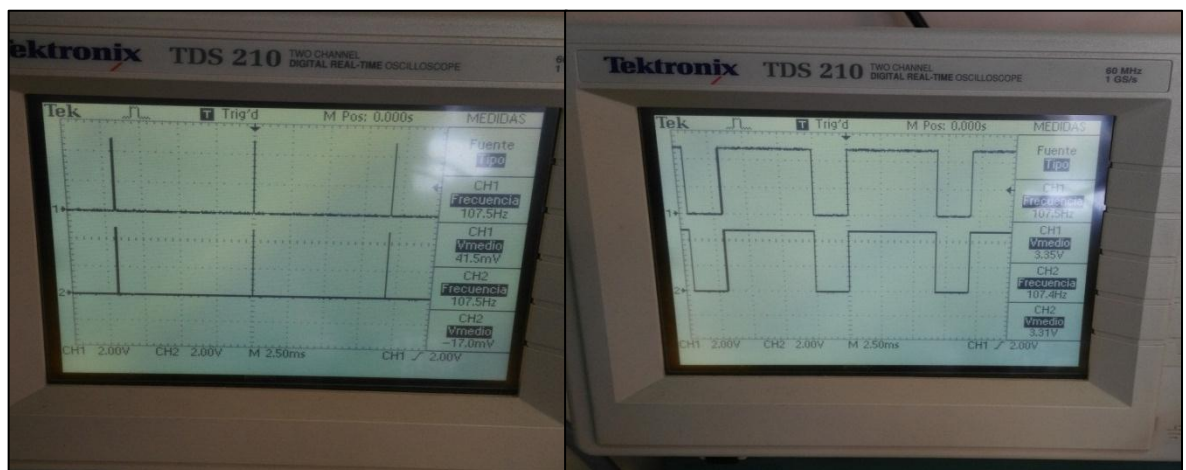


Figura 14.8- Variació amplada de polsos des de l'oscil·loscopi

Al mateix temps l'Slider 2, incrementa la tensió i la velocitat d'una roda respecte una altra a partir de la velocitat establerta en l'Slider 1 com s'observa al programa. Aquest Slider, com s'ha explicat, fa girar el vehicle modificant la velocitat d'una roda a partir de la

velocitat de l'Slider 1 si s'envia una "DADA" d'entre 101 i 201 com s'observa a la Figura 14.9.

```

if (DADA > 100 && DADA <202)
{
  if (DADA>=150)
  {
    velLdif=(DADA-150)*2;
    amplepols1 = velRecta+velLdif;

  }
  else
  {
    velRdif=(150-DADA)*2;
    amplepols2=velRecta+velRdif;
  }
}

```

Figura 14.9- Condicions de gir del programa

A la Figura 14.10 s'aprecia com l'amplitud dels dos polsos varia si s'envia una dada d'entre 101 a 201, a excepció del 150 que és el valor central del Slider.

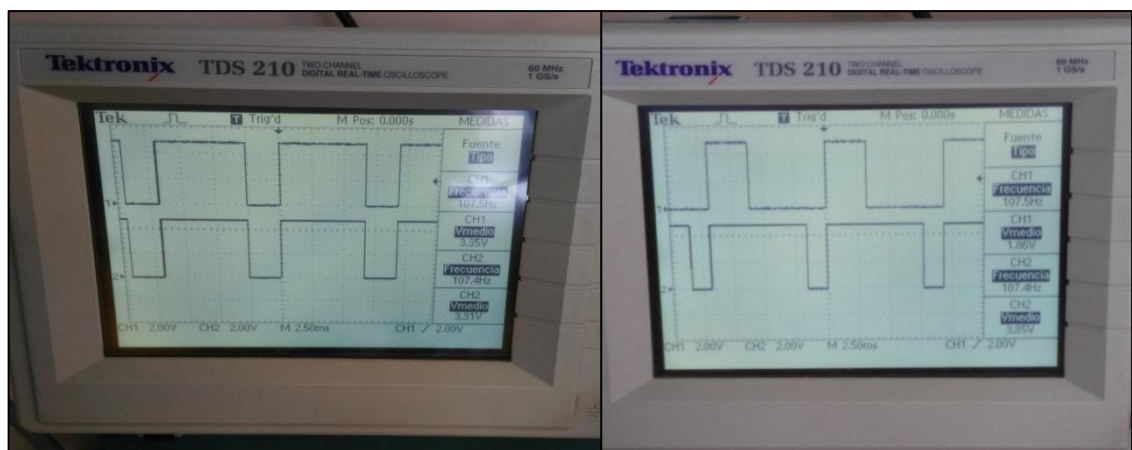


Figura 14.10- Diferenciació de l'amplada dels polsos

En els blocs de l'aplicació, primer de tot s'ha creat un bloc que el que faci sigui fer aparèixer els dos Sliders després de connectar el mòbil al dispositiu bluetooth HC-05. S'observa a la Figura 14.11 com després de connectar-se via bluettoth amb "Call

BluetoothClient1.Conectadress", s'han definit els valors màxim i mínim que envia cada Sliders i la longitud que tindran a la pantalla del mòbil.

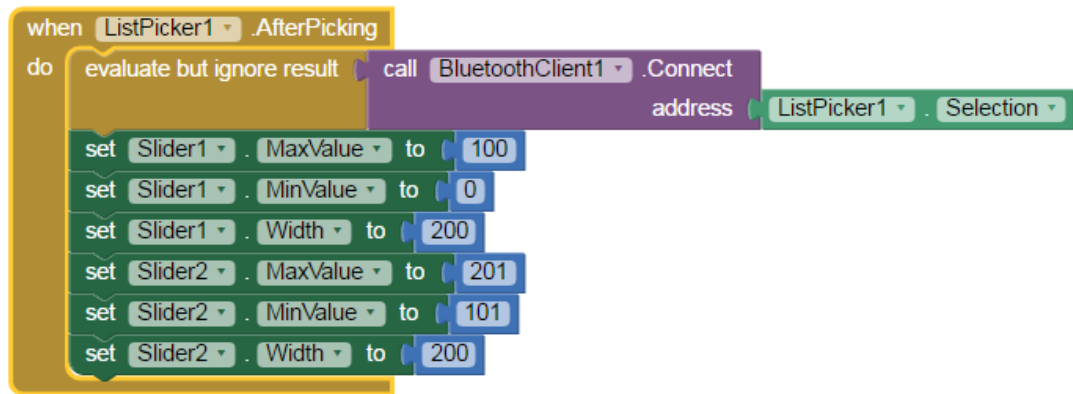


Figura 14.11- Funció per definir els Sliders

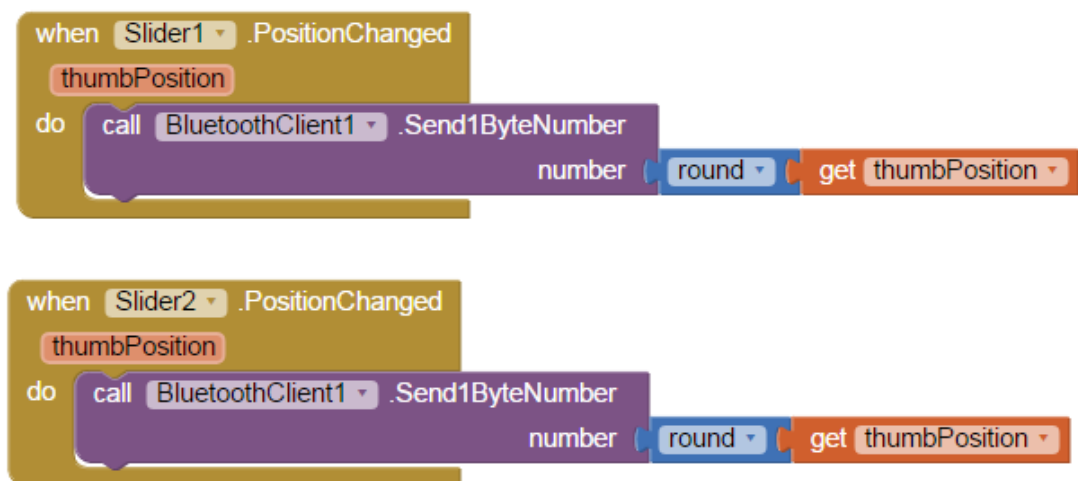


Figura 14.12- Funcions de canvi de posició de cada Slider

Després s'ha creat un bloc per a cada Slider, com a la Figura 14.12, que permeti enviar la dada via bluetooth que marca cada slider (thumbPosition) cada cop que es mogui l'Slider (When Slider.PositionChanged). S'observa que cada Slider envia un valor d'un byte de longitud.

Al definir els Sliders s'ha visualitzat la forma del tren de polsos de cada motor a l'oscil·loscopi per apreciar la diferència entre les amplituds dels polsos. Com s'ha explicat anteriorment, l'Slider 1 varia l'amplitud dels polsos dels dos motors de la mateixa forma. Com s'observen a la Figura 14.13, la Figura 14.14 i la Figura 14.15, en canvi, l'Slider 2 només varia l'amplitud d'un dels motors a partir de l'amplitud que s'ha establert amb l'Slider 1 com mostren la Figura 14.16, la Figura 14.17, la Figura 14.18 i la Figura 14.19.

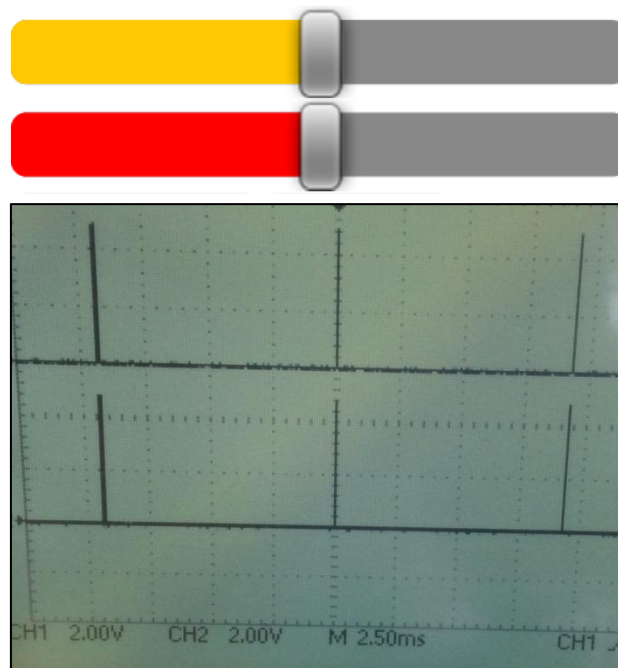


Figura 14.13- Posició inicial vehicle

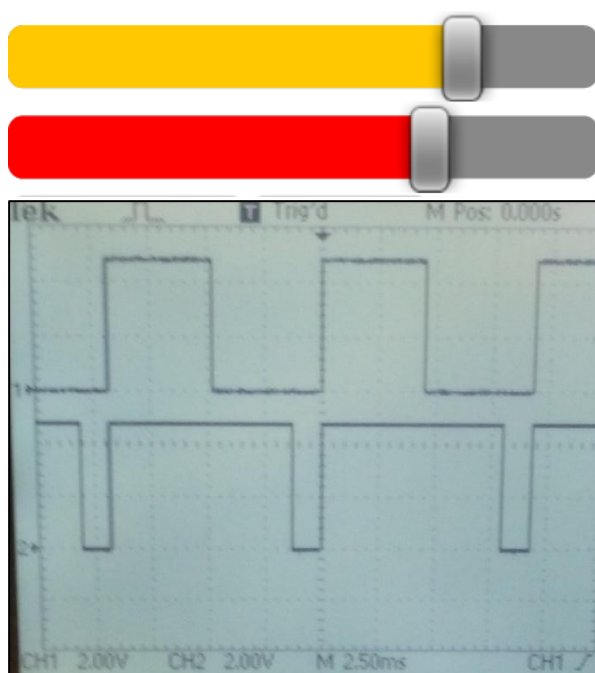


Figura 14.14- Vehicle anant recte cap endavant

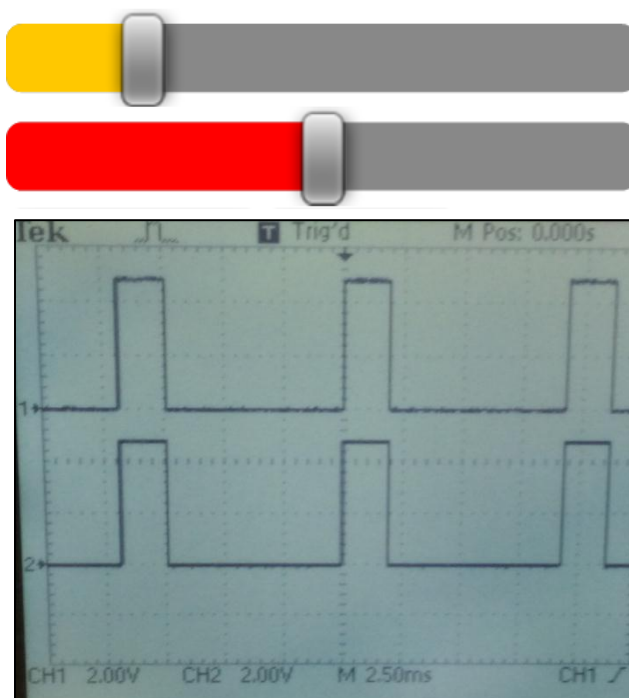


Figura 14.15- Vehicle anant recte cap endarrere

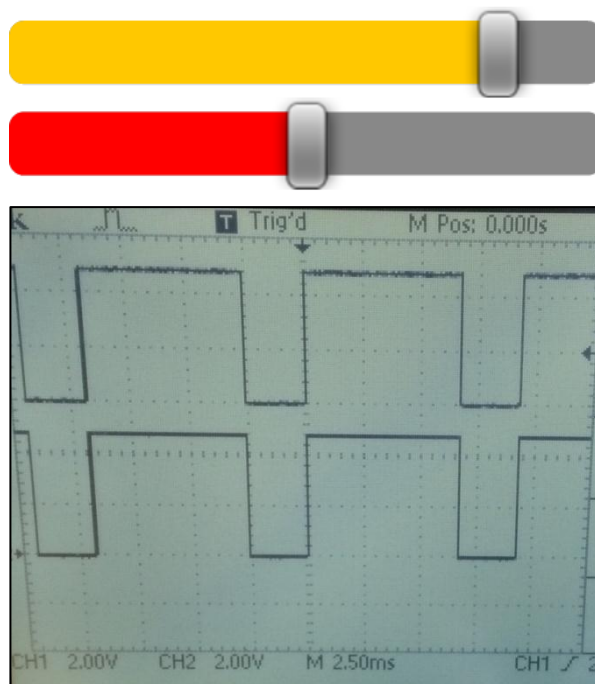


Figura 14.16- Vehicle girant a la dreta

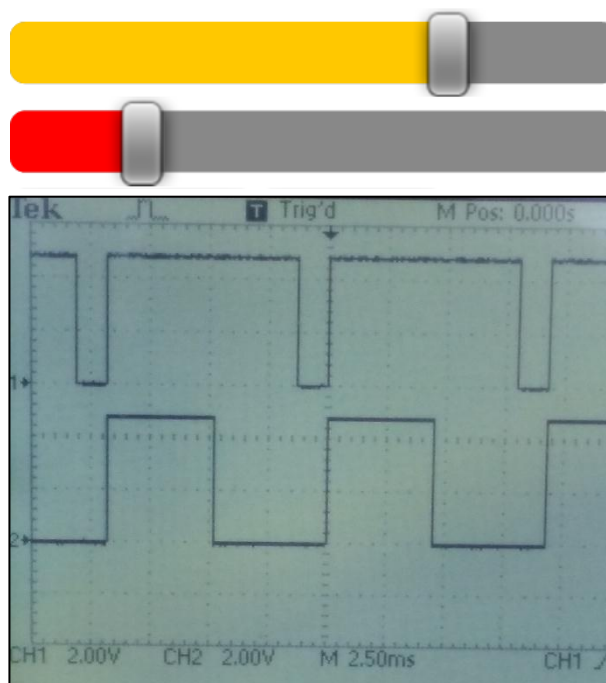


Figura 14.17- Vehicle girant a l'esquerra

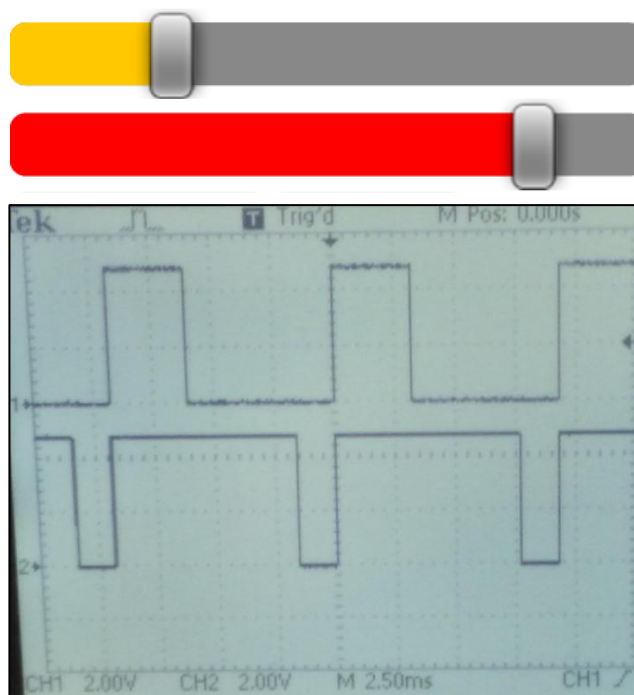


Figura 14.18- Vehicle anant endarrere i girant a l'esquerra

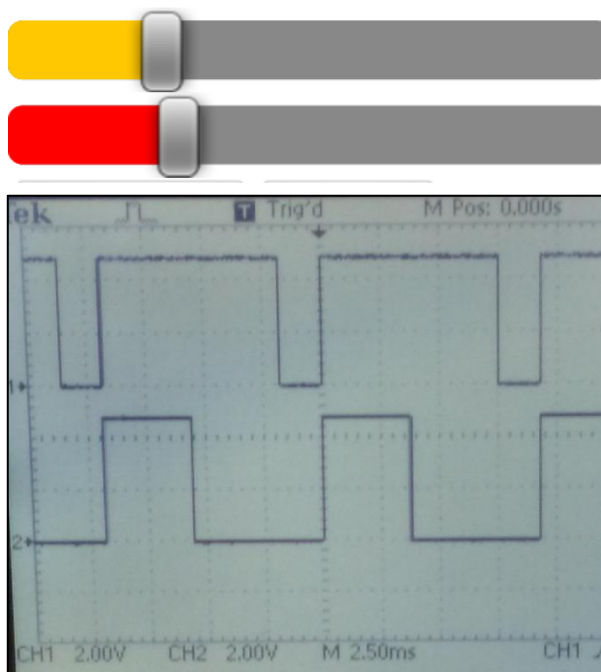


Figura 14.19- Vehicle anant endarrere i girant a la dreta

14.3.2 Botons

A part dels Sliders, s'han creat 5 botons per a controlar el vehicle més eficientment en espais reduïts, ja que el vehicle pot tenir més dificultats de moviment, sobretot al girar. En aquest aspecte, s'han creat 4 botons (GO, STOP, RIGHT, LEFT) que el codi de la Figura 14.20.

```
if (DADA==205)
{
    amplepols2 = velRecta;
    amplepols1 = velRecta;
}
if (DADA==203)
{
    amplepols1=100;
    amplepols2=0;
}
if (DADA==204)
{
    amplepols2=100;
    amplepols1=0;
}
if (DADA==206)
{
    amplepols2=0;
    amplepols1=0;
}
```

Figura 14.20- Condicions per activar els botons al programa

Els botons, el que fan és bàsicament enviar un byte d'un valor predeterminat. Si s'envia un 203 només la roda de la dreta gira, i per tant el vehicle gira a l'esquerra (LEFT) a una velocitat màxima; si s'envia un 204 el vehicle gira cap la dreta (RIGHT) amb la roda de l'esquerra girant a velocitat màxima. En canvi, si s'envia un 205 el vehicle es trasllada a la velocitat comuna velRecta cap endavant (GO) i si s'envia un 206 el vehicle s'atura (STOP).

Durant la programació va aparèixer un nou problema: Si el sensor d'ultrasons detectava un obstacle s'aturava, però després no es podia fer moure de nou. Per solucionar aquest inconvenient s'ha creat un botó anomenat Endarrere que el que fa és fer que el vehicle

vagi cap endarrere per sortir del bloqueig i poder tornar a enviar una altra dada. Aquesta maniobra segueix el codi del la Figura 14.21 que mostra com el que es fa és canviar el pin del trigger del sensor d'ultrasons HC-SR04 de sortida a entrada per a que no emeti cap senyal durant 1 s i així mentre el sensor està inhabilitat fer que el vehicle faci marxa enrere.

```

if (DADA==202)
{
  TRISBits.TRISC3=1;
  MOT=1;
  velRecta=100-(50-DADA)*2;
  amplepols2 = 5;
  amplepols1 = 5;
  __delay_ms(1000);
}

```

Figura 14.21- Botó Endarrere al programa

A la Figura 14.22 es mostra com un botó envia un byte al microcontrolador a través de la funció "When Button. Click do" que el que fa immediatament és connectar-se al bluetooth per enviar un valor definit. L'operació round és per arrodonir la possible conversió a byte del valor enviat.

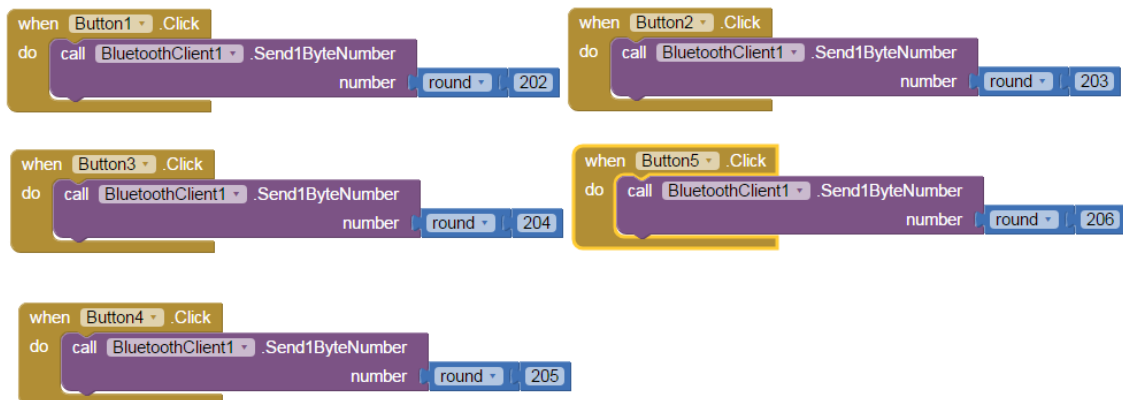


Figura 14.22- Funcions de cada botó a APP-Inventor

14.4 Dades Sensor de llum

La segona part de l'aplicació és mostrar a l'usuari les dades que capta el sensor APDS-9960 sobre el color de la llum que capta.

En aquesta part s'ha muntat una graella de 2 columnes i 4 files on s'ha creat a una caixa de text (TextBox) per a cada valor. La columna de la dreta serà pel byte LSB (Least significant Byte) i la columna de la dreta serà pel byte MSB (Most Significant Byte) de cada valor que es vulgui llegir. Els 4 valors que es llegeixen a l'aplicació són el C_Data (intensitat lluminosa), el R_Data (vermell), el G_Data(verd) i el B_Data(blau).

Per enviar les dades el PIC16F690 escriu les aquestes dades a través de la UART com es mostra a la Figura 14.23.

```
UART_Write(C_data_L);  
__delay_ms(10);  
UART_Write(C_data_H);  
__delay_ms(10);  
UART_Write(R_data_L);  
__delay_ms(10);  
UART_Write(R_data_H);  
__delay_ms(10);  
UART_Write(G_data_L);  
__delay_ms(10);  
UART_Write(G_data_H);  
__delay_ms(10);  
UART_Write(B_data_L);  
__delay_ms(10);  
UART_Write(B_data_H);  
__delay_ms(10);
```

Figura 14.23- Escriptura de les dades del sensor per la UART al programa

A l'aplicació s'ha afegit un rellotge (Clock1) que el que fa és actualitzar els valors de la taula cada 1000 ms, és a dir, cada segon.

Com s'observa a la Figura 14.24, dins dels blocs de l'aplicació s'ha creat un bloc que fa que s'escrigui a cada caps de text el seu valor corresponent que li arriba a través del bluetooth i que cada valor s'actualitzi cada segon.

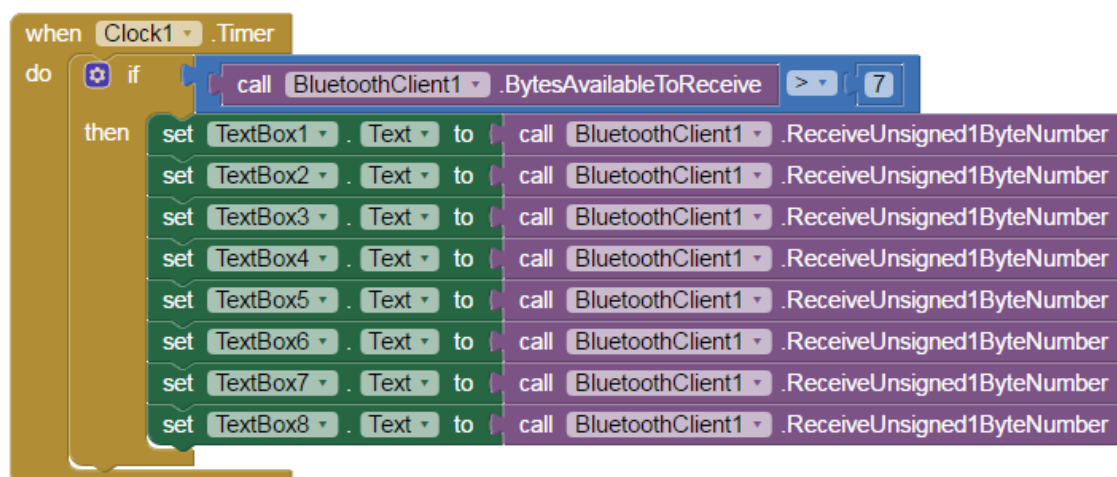


Figura 14.24- Funció de mostrar les dades de cada color al mòbil

15. Aplicació del sensor

15.1 Control del vehicle pel color

Un cop s'ha controlat els motors amb l'aplicació de mòbil s'ha decidit adjuntar la informació que capta el sensor de llum amb el moviment del vehicle. Per fer-ho s'ha decidit que depenent del color que capti, el vehicle modifiqui la seva velocitat.

Com s'ha explicat a l'apartat del sensor APDS-9960, el sensor només és capaç de donar com a informació quin color del RGB està més present en una determinada llum, per tant només es fixaran tres ordres, una per cada component dominant del color.

Primer de tot, el que s'ha fet és inicialitzar uns comptadors per cada color que augmentaran una unitat cada cop el component majoritari sigui el del seu color. És a dir, si el sensor capta el color vermell com a component dominant, el comptador del color vermell (ComptaR) augmentarà una unitat i els comptadors dels altres colors es posaran a 0. Si hi ha un canvi de color, el comptador del color vermell es torna a posar a 0 i el comptador del nou color aparegut augmenta cada vegada que el sensor capta com a majoritari el color actual.

Aquí és on entra la funcionalitat dels tres leds de color vermell, verd i blau utilitzats en el circuit. Aquesta és visualitzar a l'usuari quin és el color majoritari captat pel sensor, ja que si no, és impossible saber si el sensor de llum està captant el color esperat. Si el color majoritari és el vermell, s'encén el led vermell (Figura 15.1); si el color majoritari és el verd, s'encén el led verd (Figura 15.2); Si el color majoritari és el blau, s'encén el led blau (Figura 15.3).

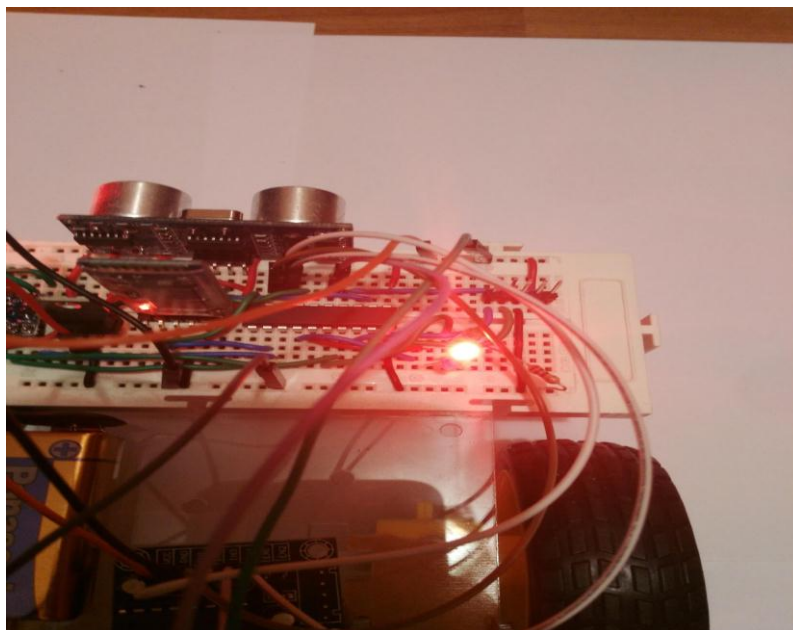


Figura 15.1- Engegada del led vermell

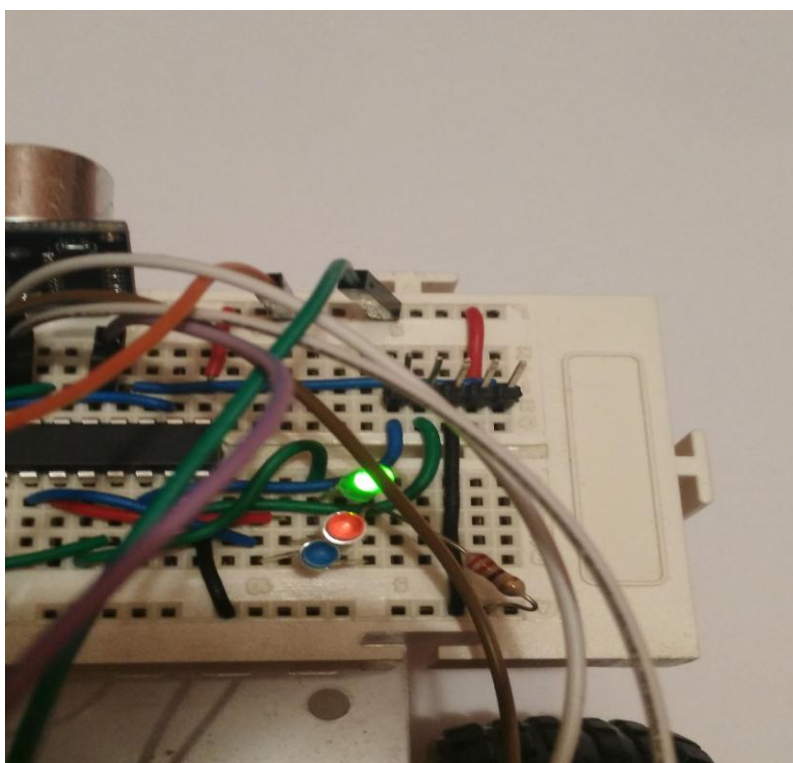


Figura 15.2- Engegada del led verd

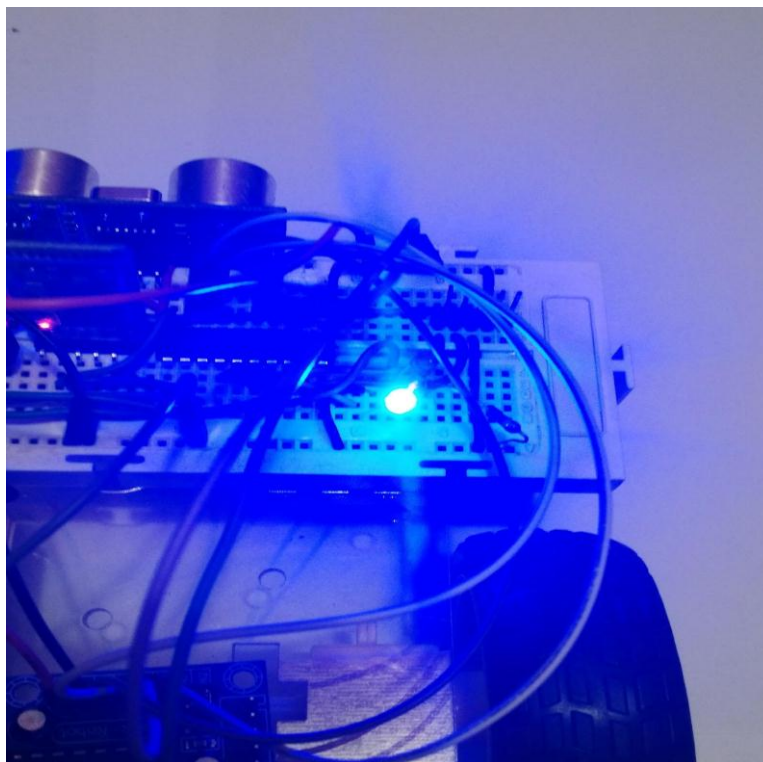


Figura 15.3- Engegada del led blau

Les ordres que realitza el vehicle són les següents:

- Si el color majoritari és el vermell, el vehicle s'atura.
- Si el color majoritari és el verd, el vehicle adopta la velocitat màxima.
- Si el color majoritari és el blau, el vehicle avança a una velocitat intermitja.


```

        if (R_data>G_data && R_data>B_data)
        {
            Red=1;
            Green=0;
            Blue=0;
            ComptaR++;
            ComptaG=0;
            ComptaB=0;

            if (ComptaR>1)
            {
                amplepols2=0;
                amplepols1=0;
                ComptaR=0; // afegit
            }
            _delay_ms(500);
            Red=0;
            Green=0;
            Blue=0;
        }
    }

```

Figura 15.4- Acció vehicle amb el vermell al programa

Quan el sensor detecta el color vermell (Figura 15.4) com el més majoritari comparant els valors R_Data, G_Data i B_data, que són els valors de 16 bits que representen la quantitat de cada color dins de la llum captada, es pot observar a la imatge com el comptador ComptaR puja de valor i com s'encén el led de color vermell que mostra quin és el color majoritari als ulls de l'usuari.

Al mateix temps hi ha una condició que quan aquest comptador sigui més gran que 1, els amplex de polsos dels dos motors i el comptador del color vermell siguin 0. Aquesta condició el que fa és aturar el vehicle i reiniciar el comptador.

```

        if (R_data>G_data && R_data>B_data)
        {
            Red=1;
            Green=0;
            Blue=0;
            ComptaR++;
            ComptaG=0;
            ComptaB=0;

            if (ComptaR>1)
            {
                amplepols2=0;
                amplepols1=0;
                ComptaR=0; // afegit
            }
        }
        __delay_ms(500);
        Red=0;
        Green=0;
        Blue=0;
    }
}

```

Figura 15.5- Acció vehicle amb el verd al programa

Quan el sensor detecta el color verd (Figura 15.5) com el més majoritari comparant els valors R_Data, G_Data i B_data, que són els valors de 16 bits que representen la quantitat de cada color dins de la llum captada, es pot observar a la imatge com el comptador ComptaG puja de valor i com s'encén el led de color verd que mostra quin és el color majoritari als ulls de l'usuari.

Al mateix temps hi ha una condició que quan aquest comptador sigui més gran que 1, els amplituds de polsos dels dos motors es posen al a 100, és a dir al màxim. Aquesta condició fa que els motors segueixin una trajectòria recta a velocitat màxima i que es reinici el comptador.

```

    if (B_data>G_data && B_data>R_data)
    {
        Blue=1;
        Red=0;
        Green=0;
        ComptaB++;
        ComptaG=0;
        ComptaR=0;

        if (ComptaB>1)
        {
            MOT=0;
            amplepols2=70; // reduir velocitat
            amplepols1=70;
            ComptaB=0; //

        }

        __delay_ms(500);
        Red=0;
        Green=0;
        Blue=0;
    }

```

Figura 15.6- Acció vehicle amb el blau al programa

Quan el sensor detecta el color blau (Figura 15.6) com el més majoritari comparant els valors R_Data, G_Data i B_data, que són els valors de 16 bits que representen la quantitat de cada color dins de la llum captada, es pot observar a la imatge com el comptador ComptaB puja de valor i com s'encén el led de color blau que mostra quin és el color majoritari als ulls de l'usuari.

Al mateix temps hi ha una condició que quan aquest comptador sigui més gran que 1, els amplituds de polsos dels dos motors es posen al a 70. Aquesta amplada ha estat programa experimentalment, ja que s'ha volgut introduir una amplada que fos capaç de fer moure el vehicle a una velocitat inferior a la màxima.

Com s'observa en les tres figures del codi, la segona part és comuna i l'únic que fa és mantenir el led encès 500 ms per agilitzar el procés de captació del color de la llum.

Un fet que s'ha d'aclarir sobre el codi és el fet que només es realitza una ordre al vehicle quan ha captat més d'una vegada un color consecutivament. Aquesta condició ve imposada perquè el sensor capta el color quan detecta una aproximació d'objecte, és a dir, quan nota un canvi superior a un llindar a la llum que rep. Si fiquessim que a la que

detecti un color realitzés la instrucció corresponent, provocaria un error a l'objectiu de controlar el vehicle imposant el color desitjat per l'usuari.

Per això s'ha imposat que a partir de la segona vegada que detecti el mateix color com a majoritari, realitzi la instrucció corresponent, ja que s'eviten errors. A més, gràcies a poder visualitzar si el color que detecta és el color real a través dels leds, dóna un cert marge de maniobra a l'usuari a la rectificació.

Per últim, només comentar que per poder controlar el vehicle s'han utilitzat els colors vermell, verd i blau, és a dir, que no s'han utilitzat barreges per a poder imposar els colors desitjats sense tenir irregularitats a l'hora de la captació de colors per part del sensor.

15.2 Relació entre color i dada enviada

En aquest apartat es vol mostrar al lector que les instruccions que imposen cada color són les mateixes que es realitzen si s'envia la dada corresponent a través de l'aplicació.

En aquest cas, només són 2 instruccions:

- Si el color que s'introdueix és el vermell, el vehicle s'atura. Aquesta acció és la mateixa que si s'envia un byte de valor 50 a través de l'aplicació del mòbil.
- Si el color que s'introdueix és el verd, el vehicle assoleix la velocitat màxima a les dues rodes. Aquesta acció és la mateixa que si s'enviés el byte 100 a través de l'aplicació de mòbil.

En el cas del color blau, com que la velocitat dels motors es selecciona a partir d'un Slider, no es pot precisar però seria el mateix que enviar el byte 85 a través de l'aplicació de mòbil.

16. Programa

El programa creat per al correcte control del vehicle està adjuntat a l'annex. La Figura 16.1 mostra quin és el diagrama de fluxe de l'algoritme que segueix el programa principal. Les figures següents mostren els diagrames de flux de les interrupcions: La figura 16.2 mostra la interrupció de la UART per a enviar dades des del telèfon mòbil, la Figura 16.3 mostra la interrupció del Timer 0 per a crear la funció dels polsos del motor, la Figura 16.4 mostra la interrupció externa RA2/INT per rebre les dades que capta el sensor i la Figura 16.5 mostra la interrupció PORTA/PORTB per a aturar el vehicle.

A l'annex també s'ha adjuntat les llibreries del laboratori d'electrònica APDS-9960.h, I2C.h i uart.h per a complementar el programa principal.

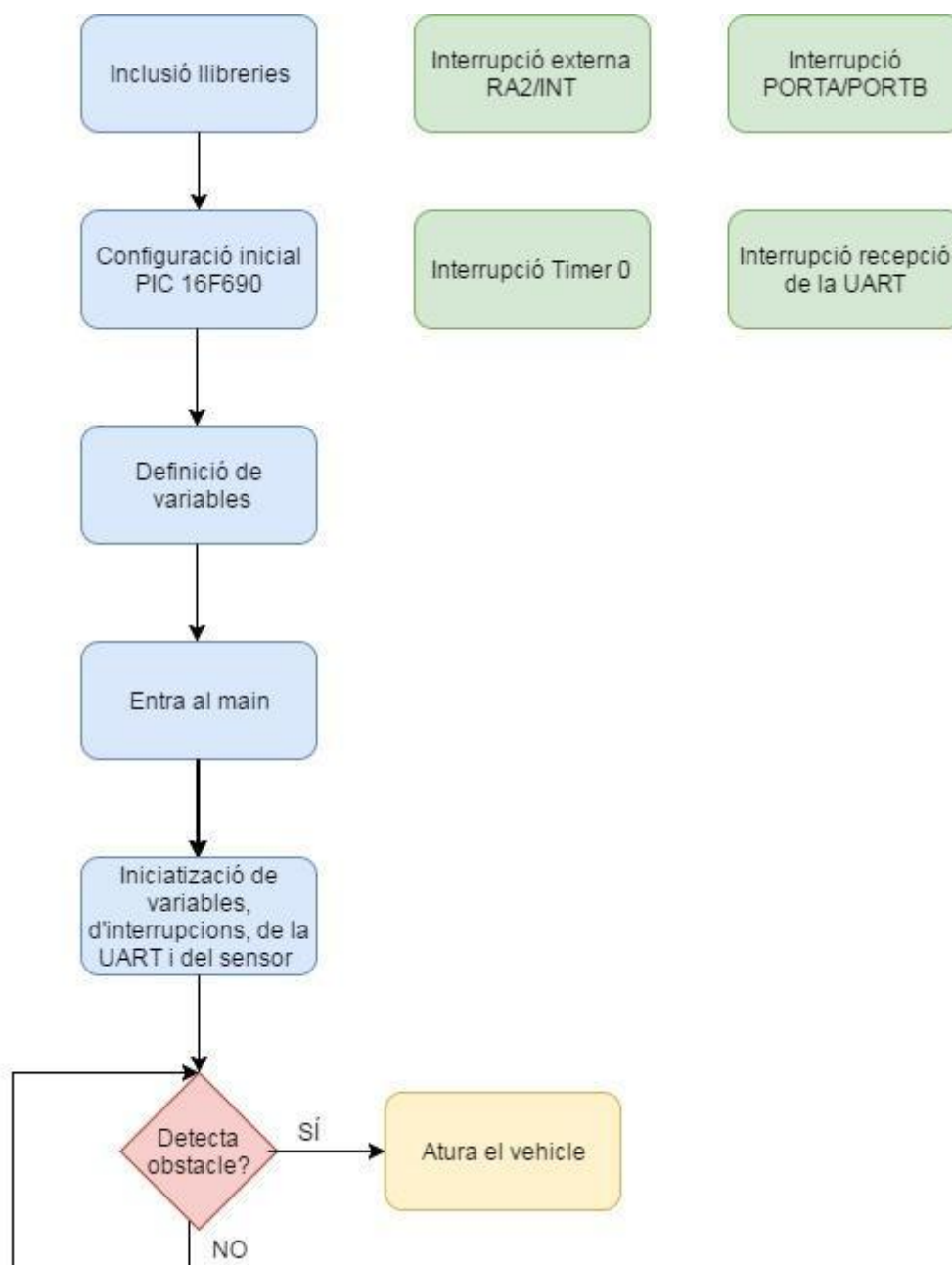


Figura 16.1- Flow Chart programa principal

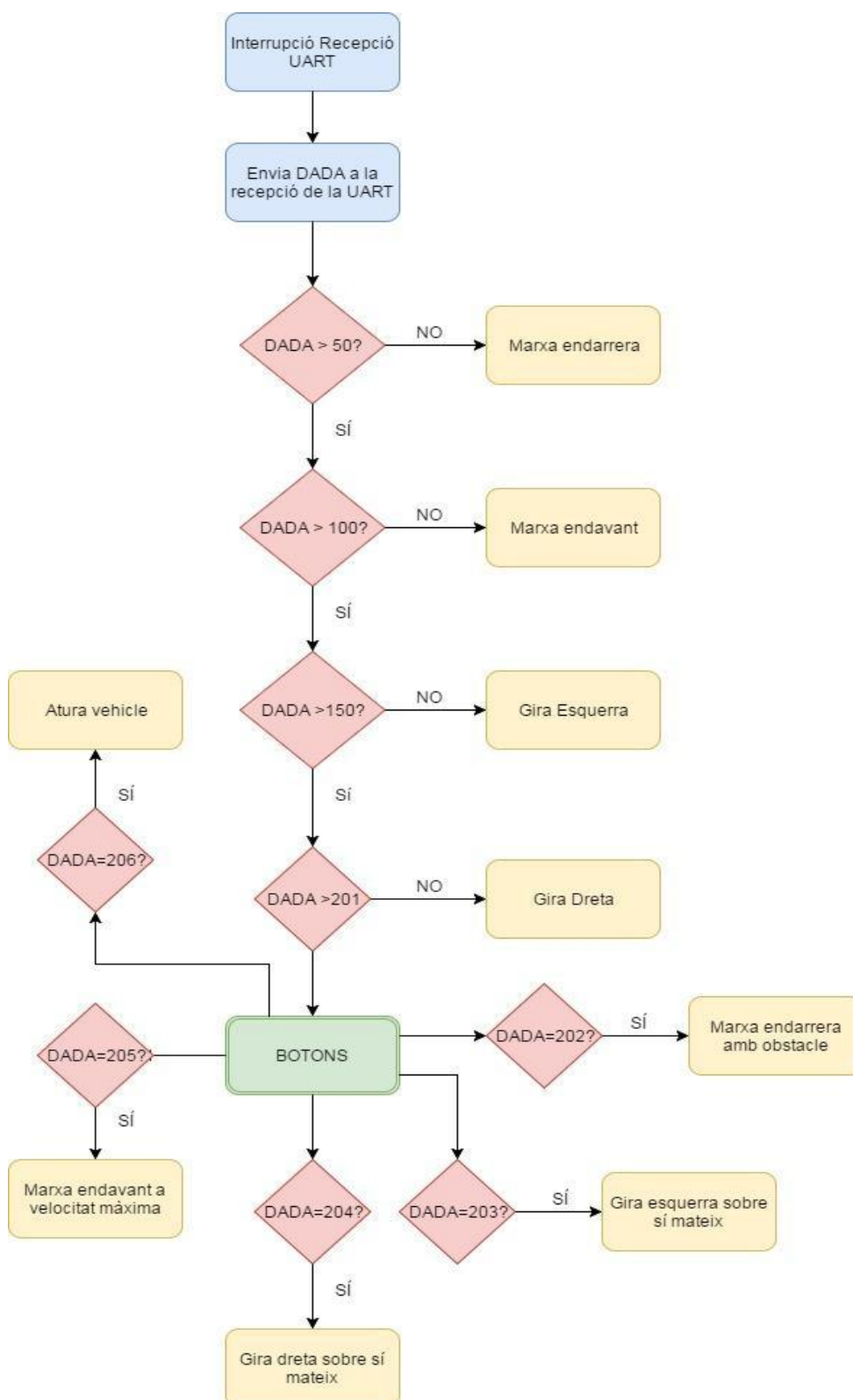


Figura 16.2- Flow Chart programa interrupció UART

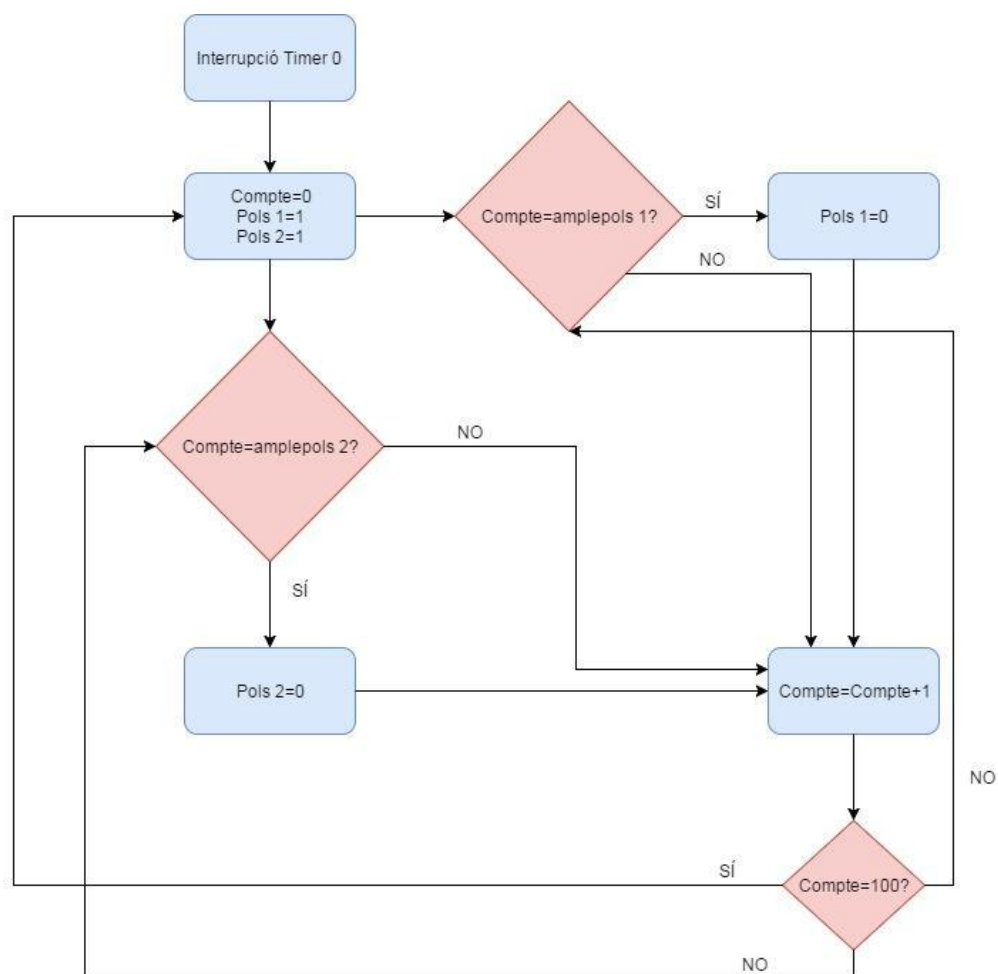


Figura 16.3- Flow Chart programa interrupció Timer 0

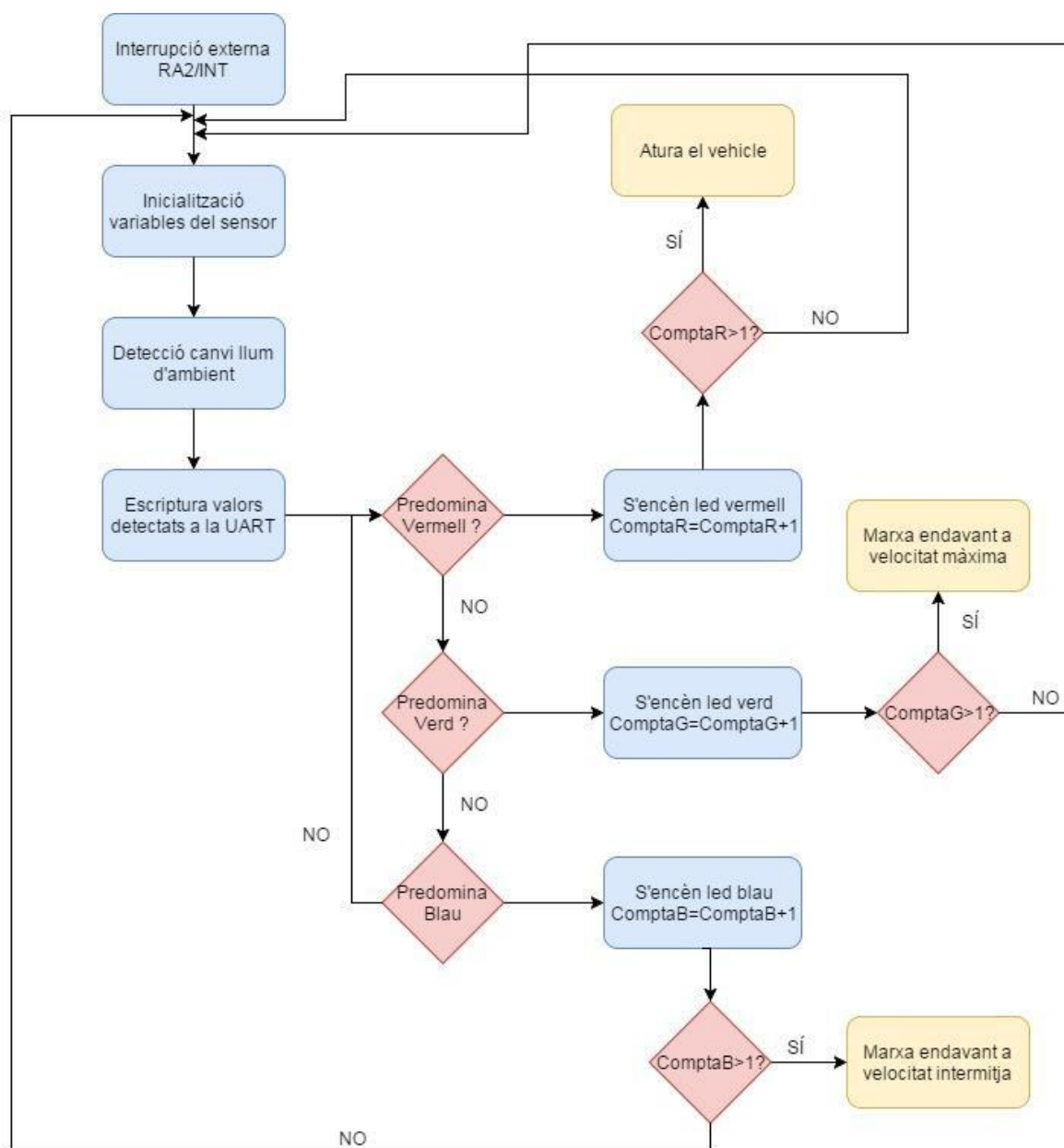


Figura 16.4- Flow Chart programa interrupció externa RA2/INT

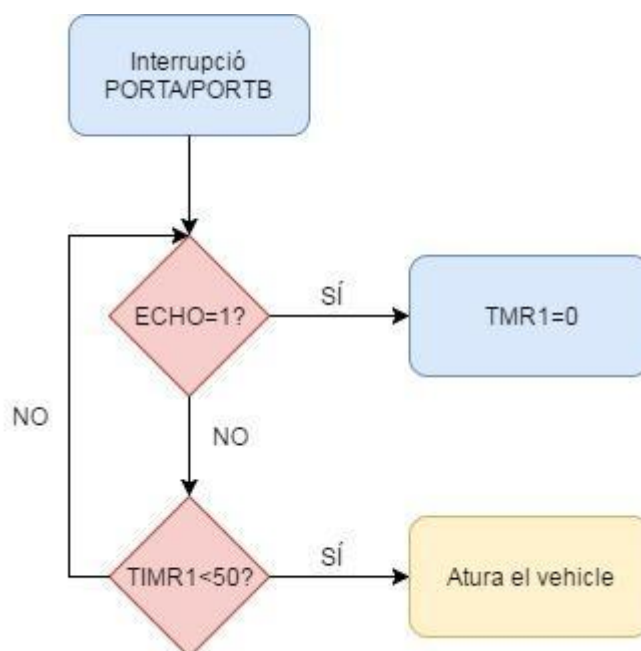


Figura 16.5- Flow Chart programa interrupció PORTA/PORTB

17. Pressupost

En aquest apartat de pressupost s'ha calculat els costos de la realització del projecte en base al material necessitat, el treball de l'enginyer junior i l'ajuda puntual del professor titulat.

Com es veu a la Taula 17.1, s'ha calculat el cost de cada material considerant el preu €/unitat i les unitats utilitzades i cost dels softwares utilitzats . En canvi a la Taula 17.2, s'ha afegit el cost del treball de l'enginyer i del professor titulat.

Element	Preu [€/unitat]	Unitats	Cost [€]
PIC 16F690	2,68	1	2,68
APDS-9960	14,5	1	14,5
HC-05	4,90	1	4,90
7805	0,31	1	0,31
1117	0,44	1	0,44
L293D	1,36	1	1,36
Pila 9V	1,50	6	9,00
PICKIT 3	36,99	1	36,99
Protoboard	7,18	1	7,18
HC-SR04	1,80	1	1,80
Pack Vehicle	42,95	1	42,95

Element	Preu [€/unitat]	Unitats	Cost [€]
Cables de coure	8,00	3	8,00
Mòbil	250	1	250
Ordinador portàtil	250	1	250
Leds	0,1	3	0,30
Resistència 220 Ω	1,19	1	1,19
MPLAB IDE	0	1	0
Compilador XC8	0	1	0
MIT app inventor	0	1	0
Canviador nivell	0,8	1	0,80
TOTAL			648,40

Taula 17.1- Costos de material

	Preu [€/h]	Temps [h]	Cost [€]
Treball			
Enginyer junior	15	300	4500
Professor titulat	50	10	500
TOTAL			5000

Figura 17.2- Costos de personal

El pressupost total del projecte és **5648,40 €** , que és la suma de tots els elements i la feina feta per part de l'enginyer junior que ha desenvolupat el projecte i el professor titulat que li ha fet el seguiment.

18. Impacte mediambiental

L'impacte mediambiental del projecte és mínim. Primer de tot, cal destacar que la majoria de components han estat comprats a fabricants externs i que per tant, només s'ha hagut d'utilitzar i en cap cas ens ha produït cap impacte la seva producció.

L'impacte principal és que com a energia de subministrament s'han utilitzat varies piles de 9 V que han estat reciclades un cop s'han gastat, és a dir, quan la tensió subministrada era de 7 V i el regulador 7805 no podia funcionar adequadament. Durant els assajos, s'ha consumit l'energia de l'ordinador portàtil per reduir el consum autònom de les piles.

Per últim cal destacar que tota la construcció del vehicle s'ha fet manualment, i que per tant l'impacte mediambiental ha estat mínim, ja que només s'han soldat alguns components com els motors, els sensor APDS-9960 i el canviador de nivell. No obstant, s'ha de destacar que s'ha llançat varis cms de cablejat de coure-estany, ja que venien embobinats per part del fabricant i s'han hagut de tallar a mida per adaptar-se a les dimensions del circuit electrònic a la protoboard.

19. Conclusions

El projecte mostra com construir un circuit electrònic per a controlar un vehicle a través d'una aplicació de mòbil i a través de la informació captada per el sensor APDS-9960. Com s'ha vist, no és suficient treballar només amb un microcontrolador PIC 16F690, sinó que calen altres elements per a controlar eficientment el vehicle com un mòdul Bluetooth HC-05 per a controlar el vehicle a través del mòbil i un sensor d'ultrasons HC-SR04 que eviti que el vehicle xoqui contra un obstacle.

A més, s'ha observat com cada dispositiu té les seves especificacions de funcionament i que s'han d'adaptar mitjançant reguladors de tensió i canviadors de nivell al conjunt del circuit per a facilitar la programació del microcontrolador. Un fet molt important del projecte és la utilització de l'amplificador de corrent L293D per a poder adaptar els senyals que ha creat el circuit electrònic amb les especificacions electromecàniques dels motors per a poder girar els seus eixos.

En la manera de programar és apreciable la versatilitat de treballar en un llenguatge com el C, a diferència de fer-ho en llenguatge d'ensamblador, ja que les instruccions realitzades al programa són més agraïdes a l'hora de seguir el programa tant per a la persona que programa com per a qualsevol usuari que no hagi treballat amb el PIC 16F690

S'ha vist una forma bastant usual de comunicació entre dos dispositius, el microcontrolador PIC 16F690 i el sensor APDS-9960, amb el protocol I2C, ja que permet que només estiguin comunicats segons la necessitat del projecte.

Finalment, s'ha demostrat com actualment, fabricar una aplicació d'Android està a l'abast de tothom, ja que existeixen programes que permeten que el dissenyador de l'aplicació pugui construir l'aplicació sense entendre de llenguatges de programació

20. Agraïments

Per realitzar un projecte, a vegades, un enginyer no dóna a l'abast i es veu obligat a buscar solucions alternatives amb l'ajuda d'altres professionals. En aquest projecte, l'autor agraeix l'ajut de la tutora Rosa Rodríguez Montañés, no només pel seu adequat seguiment i la seva alta disponibilitat a ajudar que el projecte arribés al final, sinó també perquè ha intervingut a l'adquisició de certs dispositius electrònics i ha disposat a l'autor de tot el material necessari. També s'ha d'agrair als professors del departament d'electrònica perquè han estat sempre oberts a facilitar l'espai del laboratori d'electrònica quan ha fet falta.

Finalment, agrair a la família i als companys d'estudis per haver estat sempre a disposició d'ajudar no només en aquest treball de final de grau, sinó durant tota l'etapa que es tanca amb la realització d'aquest treball.

21. Bibliografia

[1] AVAGO technologies, APDS-9960, 13 Novembre 2015

[2] Dragos Mihai, Eugen Strajescu, From Wavelength to RGB Filter, Vol 69, No 2, Bucharest, 2007

[3] MICRHOCHIP, PIC16F631/677/685/687/689/690 Data Sheet, USA, 2008

[4] MICROCHIP, PICKIT 3 Programmer and Debugger User's Guide, USA, 2010

[5] STMicroelectronics, LD1117SERIES, 2005

[6] Texas Instruments, L293D QUADRUPLE HALF-H DRIVER, USA, 1998

[7] Texas Instruments, μ A7800 SERIES POSITIVE-VOLTAGE REGULATORS, 2004

[8] TTMotor, TGP01D-A130 Gear motor

[9] APP INVENTOR

<<http://appinventor.mit.edu/explore/about-us.html>>

[10] Bluetooth Module HC-05

<<http://microcontrollerslab.com/bluetooth-module-hc-05-interfacing-pic/>>

[11] Deek Robot

<<http://www.electronicoscaldas.com/robotica/502-plataforma-para-carro-robot-2wd-tipotriciclo.html>>

[12] Interfacing HC-SR04 Ultrasonic Sensor with PIC Microcontroller

<<https://electrosome.com/hc-sr04-ultrasonic-sensor-pic/>>

[13] Plataforma movil 2WD

<<http://www.electronicoscaldas.com/robotica/502-plataforma-para-carro-robot-2wd-tipo-triciclo.html>>

[14] Wavelength to RGB and HEX Calculator

<<http://lsrtools.1apps.com/wavetorgb/index.asp?wavelength=500>>